

# Altitude™ 35x0 Access Point Product Reference Guide, Software Version 2.3



Extreme Networks, Inc.  
3585 Monroe Street  
Santa Clara, California 95051  
(888) 257-3000  
(408) 579-2800  
<http://www.extremenetworks.com>  
Published: December 2009  
Part Number: 100347-00 Rev 01



AccessAdapt, Alpine, Altitude, BlackDiamond, EPICenter, ExtremeWorks Essentials, Ethernet Everywhere, Extreme Enabled, Extreme Ethernet Everywhere, Extreme Networks, Extreme Standby Router Protocol, Extreme Turbodriven, Extreme Velocity, ExtremeWare, ExtremeWorks, ExtremeXOS, Go Purple Extreme Solution, ExtremeXOS ScreenPlay, ReachNXT, Sentiariant, ServiceWatch, Summit, SummitStack, Triumph, Unified Access Architecture, Unified Access RF Manager, UniStack, the Extreme Networks logo, the Alpine logo, the BlackDiamond logo, the Extreme Turbodriven logo, the Summit logos, and the Powered by ExtremeXOS logo are trademarks or registered trademarks of Extreme Networks, Inc. or its subsidiaries in the United States and/or other countries.

sFlow is a registered trademark of InMon Corporation.

Specifications are subject to change without notice.

All other registered trademarks, trademarks, and service marks are property of their respective owners.

© 2009 Extreme Networks, Inc. All Rights Reserved.

# Table of Contents

|   |          |
|---|----------|
| <b>Chapter 1: About This Guide.....</b>                 | <b>7</b> |
| Introduction .....                                      | 7        |
| Document Conventions .....                              | 7        |
| Notational Conventions .....                            | 7        |
| <b>Chapter 2: Introduction.....</b>                     | <b>9</b> |
| Feature Overview.....                                   | 9        |
| AP35xx Hardware SKUs.....                               | 10       |
| Dual Mode Radio Support .....                           | 10       |
| Multiple Mounting Options.....                          | 11       |
| Antenna Support for 2.4 GHz and 5 GHz Radios .....      | 11       |
| Sixteen Configurable WLANs .....                        | 11       |
| Support for 4 BSSIDs per Radio .....                    | 11       |
| Quality of Service (QoS) Support.....                   | 11       |
| Industry Leading Data Security.....                     | 12       |
| EAP Authentication .....                                | 12       |
| WEP Encryption .....                                    | 12       |
| Wi-Fi Protected Access (WPA) Using TKIP Encryption..... | 13       |
| WPA2-CCMP (802.11i) Encryption .....                    | 13       |
| Firewall Security .....                                 | 13       |
| VPN Tunnels.....  | 14       |
| Content Filtering .....                                 | 14       |
| VLAN Support.....                                       | 14       |
| Updatable Firmware .....                                | 14       |
| Programmable SNMP v1/v2/v3 Trap Support.....            | 14       |
| Power-over-Ethernet Support .....                       | 15       |
| MU-MU Transmission Disallow .....                       | 15       |
| Voice Prioritization .....                              | 15       |
| Support for CAM and PSP MUs .....                       | 15       |
| Transmit Power Control .....                            | 16       |
| Advanced Event Logging Capability.....                  | 16       |
| Configuration File Import/Export Functionality.....     | 16       |
| Default Configuration Restoration .....                 | 16       |
| DHCP Support .....                                      | 16       |
| Multi-Function LEDs .....                               | 16       |
| Mesh Networking .....                                   | 17       |
| Additional LAN Subnet .....                             | 17       |
| On-board Radius Server Authentication .....             | 18       |
| Hotspot Support.....                                    | 18       |
| Dynamic DNS .....                                       | 18       |
| Auto Negotiation .....                                  | 18       |
| IP Filtering .....                                      | 18       |
| DHCP Lease Information.....                             | 19       |
| Configurable MU Idle Timeout.....                       | 19       |
| Auto Channel Select (ACS) Smart Scan .....              | 19       |
| Trusted Host Management .....                           | 19       |

|   |           |
|---|-----------|
| Rogue AP Enhancements .....                             | 19        |
| Radius Time-Based Authentication .....                  | 19        |
| QBSS Support .....                                      | 20        |
| Reliable Multicast .....                                | 20        |
| Configurable WPA Handshake Retry Levels .....           | 20        |
| Theory of Operations .....                              | 20        |
| Cellular Coverage .....                                 | 21        |
| MAC Layer Bridging.....                                 | 21        |
| Media Types .....                                       | 22        |
| MU Association Process.....                             | 22        |
| <b>Chapter 3: CLI Reference .....</b>                   | <b>25</b> |
| Connecting to the CLI .....                             | 25        |
| Accessing the CLI through the Serial Port .....         | 25        |
| Accessing the CLI via Telnet .....                      | 26        |
| Admin and Common Commands .....                         | 27        |
| Network Commands .....                                  | 35        |
| Network LAN Commands .....                              | 36        |
| Network LAN, Bridge Commands .....                      | 39        |
| Network LAN, WLAN-Mapping Commands .....                | 42        |
| Network LAN, DHCP Commands .....                        | 50        |
| Network Type Filter Commands .....                      | 56        |
| Network WAN Commands .....                              | 61        |
| Network WAN NAT Commands .....                          | 64        |
| Network WAN, VPN Commands .....                         | 70        |
| Network WAN Content Commands .....                      | 79        |
| Network WAN, Dynamic DNS Commands.....                  | 83        |
| Network Wireless Commands.....                          | 87        |
| Network WLAN Commands.....                              | 88        |
| Network Security Commands .....                         | 94        |
| Network Security Policy Edit Commands.....              | 100       |
| Network ACL Commands.....                               | 107       |
| Network Radio Configuration Commands .....              | 112       |
| Network Quality of Service (QoS) Commands.....          | 127       |
| Network Bandwidth Management Commands.....              | 132       |
| Network Rogue-AP Commands.....                          | 135       |
| WIPS Commands .....                                     | 145       |
| Network MU Locationing Commands .....                   | 148       |
| Network Reliable Multicast Commands.....                | 151       |
| Network DOT 11i Retry Commands .....                    | 156       |
| Network Firewall Commands .....                         | 159       |
| Network Router Commands .....                           | 164       |
| System Commands.....                                    | 170       |
| Adaptive AP Setup Commands .....                        | 176       |
| System Access Commands .....                            | 180       |
| System Certificate Management Commands.....             | 183       |
| System SNMP Commands.....                               | 196       |
| System SNMP Access Commands .....                       | 197       |
| System SNMP Traps Commands .....                        | 202       |
| System User Database Commands .....                     | 208       |
| Adding and Removing Users from the User Database .....  | 209       |
| Adding and Removing Groups from the User Database ..... | 214       |

|   |            |
|---|------------|
| System Radius Commands .....  | 221        |
| System Network Time Protocol (NTP) Commands .....                         | 244        |
| System Log Commands .....   | 249        |
| System Configuration-Update Commands .....                                | 255        |
| Firmware Update Commands .....  | 262        |
| Statistics Commands .....   | 266        |
| <b>Chapter 4: AP Management From Controller .....</b>                     | <b>283</b> |
| Where to Go From Here .....   | 283        |
| AP Management .....   | 284        |
| Types of Adopted APs .....  | 284        |
| Licensing .....   | 284        |
| Controller Discovery .....  | 284        |
| Auto Discovery using DHCP .....   | 285        |
| Securing a Configuration Channel Between Controller and AP .....          | 285        |
| AP WLAN Topology .....  | 286        |
| Configuration Updates .....   | 286        |
| Securing Data Tunnels between the Controller and AP .....                 | 286        |
| Managing an AP's Controller Failure .....                                 | 287        |
| Remote Site Survivability (RSS) .....                                     | 287        |
| Mesh Support .....  | 287        |
| AP Radius Proxy Support .....   | 287        |
| Supported AP Topologies .....   | 288        |
| Topology Deployment Considerations .....                                  | 288        |
| Extended WLANs Only .....   | 289        |
| Independent WLANs Only .....  | 289        |
| Extended WLANs with Independent WLANs .....                               | 289        |
| Extended VLAN with Mesh Networking .....                                  | 290        |
| How the AP Receives its Configuration .....                               | 291        |
| AP Adoption Prerequisites .....   | 291        |
| Configuring the AP for Adoption by the Controller .....                   | 291        |
| Configuring the Controller for AP Adoption .....                          | 292        |
| Establishing Controller Managed AP Connectivity .....                     | 292        |
| AP Configuration .....  | 292        |
| Adopting an AP Using a Configuration File .....                           | 293        |
| Adopting an AP Using DHCP Options .....                                   | 293        |
| Controller Configuration .....  | 293        |
| AP Deployment Considerations .....  | 297        |
| Sample Controller Configuration File for IPsec and Independent WLAN ..... | 298        |
| <b>Appendix A: Country Codes .....</b>                                    | <b>303</b> |
| <b>Appendix B: Customer Support .....</b>                                 | <b>307</b> |
| Registration .....  | 307        |
| Documentation .....   | 307        |



# 1 About This Guide

## Introduction

This guide provides configuration and setup information for the Extreme Networks® Altitude™ 3510 Access Point and Altitude 3550 Access Point.

For the purposes of this guide, the devices will be called the generic term “access point” when identical configuration activities are applied to both models. When *command line interface* (CLI) commands are displayed, and apply to both models, a “35xx” convention is used.

## Document Conventions

The following document conventions are used in this document:



### NOTE

---

*Indicate tips or special requirements.*



### CAUTION

---

*Indicates conditions that can cause equipment damage or data loss.*



### WARNING!

---

*Indicates a condition or procedure that could result in personal injury or equipment damage.*

## Notational Conventions

The following notational conventions are used in this document:

- Italics are used to highlight specific items in the general text, and to identify chapters and sections in this and related documents.
- Bullets (●) indicate:
  - action items
  - lists of alternatives
  - lists of required steps that are not necessarily sequential
- Sequential lists (those describing step-by-step procedures) appear as numbered lists.





The *access point* (AP) provides a bridge between Ethernet wired LANs and wireless networks. It provides connectivity between Ethernet wired networks and radio-equipped mobile units (MUs). MUs include the full line of terminals, adapters (PC cards, Compact Flash cards and PCI adapters) and other devices.

The access point provides a maximum 54Mbps data transfer rate via each radio. It monitors Ethernet traffic and forwards appropriate Ethernet messages to MUs over the network. It also monitors MU radio traffic and forwards MU packets to the Ethernet LAN.

The management of an adopted AP is conducted by the controller, once the AP connects to an Extreme Networks Summit WM3600 or Summit WM3700 wireless LAN controller and receives its configuration. For more information on AP controller management, see [“AP Management From Controller”](#).

The AP3550 is constructed to support outdoor installations, while the AP3510 model is constructed primarily for indoor deployments. An AP3550 cannot be powered by a standard 802.3af power supply and, therefore, is recommended to use with the AP3550 Power Tap designed specifically for outdoor deployments. An AP3550 model access point also must use an RJ-45 to Serial cable to establish a serial connection to a host computer.

If you are new to using an access point for managing your network, refer to [“Theory of Operations”](#) for an overview on wireless networking fundamentals.

## Feature Overview

When managed by an Extreme Networks Summit WM3600 or WM3700 WLAN controller, an AP3510 and an AP3550 support the following features:

- [“AP35xx Hardware SKUs”](#)
- [“Dual Mode Radio Support”](#)
- [“Multiple Mounting Options”](#)
- [“Antenna Support for 2.4 GHz and 5 GHz Radios”](#)
- [“Sixteen Configurable WLANs”](#)
- [“Support for 4 BSSIDs per Radio”](#)
- [“Quality of Service \(QoS\) Support”](#)
- [“Industry Leading Data Security”](#)
- [“VLAN Support”](#)
- [“Updatable Firmware”](#)
- [“Programmable SNMP v1/v2/v3 Trap Support”](#)
- [“Power-over-Ethernet Support”](#)
- [“MU-MU Transmission Disallow”](#)
- [“Voice Prioritization”](#)
- [“Support for CAM and PSP MUs”](#)

- “Transmit Power Control”
- “Advanced Event Logging Capability”
- “Configuration File Import/Export Functionality”
- “Default Configuration Restoration”
- “DHCP Support”
- “Multi-Function LEDs”
- “Mesh Networking”
- “Additional LAN Subnet”
- “On-board Radius Server Authentication”
- “Hotspot Support”
- “Dynamic DNS”
- “Auto Negotiation”
- “Feature Overview”
- “DHCP Lease Information”
- “Configurable MU Idle Timeout”
- “Auto Channel Select (ACS) Smart Scan”
- “Trusted Host Management”
- “Rogue AP Enhancements”
- “Radius Time-Based Authentication”
- “QBSS Support”
- “Reliable Multicast”
- “Configurable WPA Handshake Retry Levels”

## AP35xx Hardware SKUs

There are several hardware SKUs in the AP35xx access point family to address regional regulatory compliance:

- AP3510-US: for the United States
- AP3510-IL: for Israel
- AP3510-ROW: for rest of the world
- AP3550-US: for the United States
- AP3550-ROW: for rest of the world

## Dual Mode Radio Support

For the hardware SKUs of AP35xx-US and AP35xx-ROW, the access point is manufactured as a dual-radio model, enabling you to configure one radio for 802.11a support, and the other for 802.11b/g support.

For the AP3510-IL SKU, only the 802.11bg radio is supported due to the local regulatory restrictions.

## Multiple Mounting Options

The access point rests on a flat surface, attaches to a wall, mounts under a ceiling or above a ceiling (attic). Choose a mounting option based on the physical environment of the coverage area. Do not mount the access point in a location that has not been approved in either an AP3510 or outdoor AP3550 radio coverage site survey.

## Antenna Support for 2.4 GHz and 5 GHz Radios

The access point supports several 802.11b/g and or 802.11a radio antennas depending on the AP hardware SKU. Select the antenna best suited to the radio transmission requirements of your coverage area.

## Sixteen Configurable WLANs

A *Wireless Local Area Network (WLAN)* is a data-communications system that flexibly extends the functionalities of a wired LAN. A WLAN does not require lining up devices for line-of-sight transmission, and are thus, desirable for wireless networking. Roaming users can be handed off from one access point to another like a cellular phone system. WLANs can therefore be configured around the needs of specific groups of users, even when they are not in physical proximity. Sixteen WLANs are configurable on each access point. Each WLAN is mapped with an ESSID.

## Support for 4 BSSIDs per Radio

The access point supports four BSSIDs per radio. Each BSSID has a corresponding MAC address. The first MAC address corresponds to BSSID #1. The MAC addresses for the other three BSSIDs (BSSIDs #2, #3, #4) are derived by adding 1, 2, 3, respectively, to the radio MAC address. Multiple ESSIDs (WLANs) per BSSID is supported.

## Quality of Service (QoS) Support

The QoS implementation provides applications running on different wireless devices a variety of priority levels to transmit data to and from the access point. Equal data transmission priority is fine for data traffic from applications such as Web browsers, file transfers or email, but is inadequate for multimedia applications.

*Voice over Internet Protocol (VoIP)*, video streaming and interactive gaming are highly sensitive to latency increases and throughput reductions. These forms of higher priority data traffic can significantly benefit from the QoS implementation. The *WiFi Multimedia QoS Extensions (WMM)* implementation used by the access point shortens the time between transmitting higher priority data traffic and is thus desirable for multimedia applications. In addition, U-APSD (WMM Power Save) is also supported.

WMM defines four access categories—*voice*, *video*, *best effort* and *background*—to prioritize traffic for enhanced multimedia support.

## Industry Leading Data Security

The access point supports numerous encryption and authentication techniques to protect the data transmitting on the WLAN.

The following authentication techniques are supported:

- “EAP Authentication” The following encryption techniques are supported:
- “WEP Encryption”
- “Wi-Fi Protected Access (WPA) Using TKIP Encryption”
- “WPA2-CCMP (802.11i) Encryption”

In addition, the access point supports the following additional security features:

- “Firewall Security”
- “VPN Tunnels”
- “Content Filtering”

### EAP Authentication

The *Extensible Authentication Protocol (EAP)* feature provides access points and their associated MU's an additional measure of security for data transmitted over the wireless network. Using EAP, authentication between devices is achieved through the exchange and verification of certificates.

EAP is a mutual authentication method whereby both the MU and AP are required to prove their identities. Like Kerberos, the user loses device authentication if the server cannot provide proof of device identification.

Using EAP, a user requests connection to a WLAN through the access point. The access point then requests the identity of the user and transmits that identity to an authentication server. The server prompts the AP for proof of identity (supplied to the user) and then transmits the user data back to the server to complete the authentication process.

An MU is not able to access the network if not authenticated. When configured for EAP support, the access point displays the MU as an EAP station.

EAP is only supported on mobile devices running Windows XP, Windows 2000 (using Service Pack #4) and Windows Mobile 2003. Refer to the system administrator for information on configuring a Radius Server for EAP (802.1x) support.

### WEP Encryption

All WLAN devices face possible information theft. Theft occurs when an unauthorized user eavesdrops to obtain information illegally. The absence of a physical connection makes wireless links particularly vulnerable to this form of theft. Most forms of WLAN security rely on encryption to various extents. Encryption entails scrambling and coding information, typically with mathematical formulas called *algorithms*, before the information is transmitted. An algorithm is a set of instructions or formula for scrambling the data. A *key* is the specific code used by the algorithm to encrypt or decrypt the data. *Decryption* is the decoding and unscrambling of received encrypted data.

The same device, host computer or front-end processor, usually performs both encryption and decryption. The transmit or receive direction determines whether the encryption or decryption function

is performed. The device takes plain text, encrypts or scrambles the text typically by mathematically combining the key with the plain text as instructed by the algorithm, then transmits the data over the network. At the receiving end, another device takes the encrypted text and decrypts, or unscrambles, the text revealing the original message. An unauthorized user can know the algorithm, but cannot interpret the encrypted data without the appropriate key. Only the sender and receiver of the transmitted data know the key.

*Wired Equivalent Privacy (WEP)* is an encryption security protocol specified in the IEEE Wireless Fidelity (Wi-Fi) standard, 802.11b and supported by the AP. WEP encryption is designed to provide a WLAN with a level of security and privacy comparable to that of a wired LAN. The level of protection provided by WEP encryption is determined by the encryption key length and algorithm. An encryption key is a string of case sensitive characters used to encrypt and decrypt data packets transmitted between a mobile unit (MU) and the access point. An access point and its associated wireless clients must use the same encryption key (typically 1 through 4) to interoperate.

## Wi-Fi Protected Access (WPA) Using TKIP Encryption

*Wi-Fi Protected Access (WPA)* is a security standard for systems operating with a Wi-Fi wireless connection. WEP's lack of user authentication mechanisms is addressed by WPA. Compared to WEP, WPA provides superior data encryption and user authentication.

WPA addresses the weaknesses of WEP by including:

- a per-packet key mixing function
- a message integrity check
- an extended initialization vector with sequencing rules
- a re-keying mechanism

WPA uses an encryption method called *Temporal Key Integrity Protocol (TKIP)*. WPA employs 802.1X and *Extensible Authentication Protocol (EAP)*.

## WPA2-CCMP (802.11i) Encryption

WPA2 is a newer 802.11i standard that provides even stronger wireless security than *Wi-Fi Protected Access (WPA)* and WEP. *Counter-mode/CBC-MAC Protocol (CCMP)* is the security standard used by the *Advanced Encryption Standard (AES)*. AES serves the same function TKIP does for WPA-TKIP. CCMP computes a *Message Integrity Check (MIC)* using the proven *Cipher Block Message Authentication Code (CBC-MAC)* technique. Changing just one bit in a message produces a totally different result.

WPA2-CCMP is based on the concept of a *Robust Security Network (RSN)*, which defines a hierarchy of keys with a limited lifetime (similar to TKIP). Like TKIP, the keys the administrator provides are used to derive other keys. Messages are encrypted using a 128-bit secret key and a 128-bit block of data. The end result is an encryption scheme as secure as any the provides.

## Firewall Security

A firewall keeps personal data in and hackers out. The access point's firewall prevents suspicious Internet traffic from proliferating the access point managed network. The access point performs *Network Address Translation (NAT)* on packets passing to and from the WAN port. This combination provides enhanced security by monitoring communication with the wired network.

## VPN Tunnels

*Virtual Private Networks (VPNs)* are IP-based networks using encryption and tunneling providing users remote access to a secure LAN. In essence, the trust relationship is extended from one LAN across the public network to another LAN, without sacrificing security. A VPN behaves like a private network; however, because the data travels through the public network, it needs several layers of security. The access point can function as a robust VPN gateway.

## Content Filtering

Content filtering allows system administrators to block specific commands and URL extensions from going out through the WAN port. Therefore, content filtering affords system administrators selective control on the content proliferating the network and is a powerful screening tool. Content filtering allows the blocking of up to 10 files or URL extensions and allows blocking of specific outbound HTTP, SMTP, and FTP requests.

## VLAN Support

A *Virtual Local Area Network (VLAN)* can electronically separate data on the same AP from a single broadcast domain into separate broadcast domains. By using a VLAN, you can group by logical function instead of physical location. There are 16 VLANs supported on the access point. An administrator can map up to 16 WLANs to 16 VLANs and enable or disable dynamic VLAN assignment. In addition to these 16 VLANs, the access point supports dynamic, user-based, VLANs when using EAP authentication.

VLANs enable organizations to share network resources in various network segments within large areas (airports, shopping malls, etc.). A VLAN is a group of clients with a common set of requirements independent of their physical location. VLANs have the same attributes as physical LANs, but they enable administrators to group clients even when they are not members of the same network segment.

## Updatable Firmware

Extreme Networks periodically releases updated versions of device firmware. Extreme Networks recommends updating the access point to the latest firmware version for full feature functionality.

## Programmable SNMP v1/v2/v3 Trap Support

*Simple Network Management Protocol (SNMP)* facilitates the exchange of management information between network devices. SNMP uses *Management Information Bases (MIBs)* to manage the device configuration and monitor Internet devices in remote locations. MIB information accessed via SNMP is defined by a set of managed objects called *Object Identifiers (OIDs)*. An object identifier (OID) is used to uniquely identify each object variable of a MIB.

SNMP allows a network administrator to configure the access point, manage network performance, find and solve network problems, and plan for network growth. The access point supports SNMP management functions for gathering information from its network components.

The access point's SNMP agent functions as a command responder and is a multilingual agent responding to SNMPv1, v2c and v3 managers (command generators). The factory default configuration maintains SNMPv1/2c support of community names, thus providing backward compatibility.

## Power-over-Ethernet Support

When users purchase an Extreme Networks WLAN solution, they often need to place access points in obscure locations. In the past, a dedicated power source was required for each access point in addition to the Ethernet infrastructure. This often required an electrical contractor to install power drops at each access point location.

An approved power injector solution merges power and Ethernet into one cable, reducing the burden of installation and allows optimal access point placement in respect to the intended radio coverage area.

The AP3510 Power Injector is a single-port, 802.3af compliant Power over Ethernet hub combining low-voltage DC with Ethernet data in a single cable connecting to the access point. The Power Injector's single DC and Ethernet data cable creates a modified Ethernet cabling environment on the access point's LAN port eliminating the need for separate Ethernet and power cables.

The AP3550 Power Tap is a single-port Power over Ethernet hub combining low-voltage DC with Ethernet data in a single cable connecting to the access point. However, the Power Tap is specifically designed and ruggedized for use with an AP3550's outdoor deployment.

## MU-MU Transmission Disallow

The access point's MU-MU Disallow feature prohibits MUs from communicating with each other even if on the same WLAN, assuming one WLAN is configured to disallow MU-MU communication. Therefore, if an MU's WLAN is configured for MU-MU disallow, it will not be able to communicate with any other MUs connected to this access point.

## Voice Prioritization

Each access point WLAN has the capability of having its QoS policy configured to prioritize the network traffic requirements for associated MUs. A WLAN QoS page is available for each enabled WLAN on both the 802.11a and 802.11b/g radio.

Use the QoS page to enable voice prioritization for devices to receive the transmission priority they may not normally receive over other data traffic. Voice prioritization allows the access point to assign priority to voice traffic over data traffic, and (if necessary) assign legacy voice supported devices (non WMM supported voice devices) additional priority.

## Support for CAM and PSP MUs

The access point supports both CAM and PSP powered MUs. *CAM (Continuously Aware Mode)* MUs leave their radios on continuously to hear every beacon and message transmitted. These systems operate without any adjustments by the access point.

A beacon is a uniframe system packet broadcast by the AP to keep the network synchronized. A beacon includes the ESSID, MAC address, Broadcast destination addresses, a time stamp, a *DTIM (Delivery Traffic Indication Message)* and the *TIM (Traffic Indication Map)*.

*PSP (Power Save Polling)* MUs power off their radios for short periods. When a MU in PSP mode associates with an access point, it notifies the access point of its activity status. The access point responds by buffering packets received for the MU. PSP mode is used to extend an MU's battery life by enabling the MU to "sleep" during periods of inactivity.



## Transmit Power Control

The access point has a configurable power level for each radio. This enables the network administrator to define the antenna's transmission power level in respect to the access point's placement or network requirements as defined in the site survey.

## Advanced Event Logging Capability

The access point provides the capability for periodically logging system events. Logging events is useful in assessing the throughput and performance of the access point or troubleshooting problems on the access point managed Local Area Network (LAN).

## Configuration File Import/Export Functionality

Configuration settings for an access point can be downloaded from the current configuration of another access point. This affords the administrator the opportunity to save the current configuration before making significant changes or restoring the default configuration.

## Default Configuration Restoration

The access point has the ability to restore its default configuration or a partial default configuration (with the exception of current WAN and SNMP settings). Restoring the default configuration is a good way to create new WLANs if the MUs the access point supports have been moved to different radio coverage areas.

## DHCP Support

The access point can use *Dynamic Host Configuration Protocol (DHCP)* to obtain a leased IP address and configuration information from a remote server. DHCP is based on the BOOTP protocol and can coexist or interoperate with BOOTP. Configure the access point to send out a *DHCP request* searching for a *DHCP/BOOTP* server to acquire HTML, firmware or network configuration files when the access point boots. Because BOOTP and DHCP interoperate, whichever responds first becomes the server that allocates information.

The access point can be set to only accept replies from DHCP or BOOTP servers or both (this is the default setting). Disabling DHCP disables BOOTP and DHCP and requires network settings to be set manually. If running both DHCP and BOOTP, do not select BOOTP Only. BOOTP should only be used when the server is running BOOTP exclusively.

The DHCP client automatically sends a DHCP request at an interval specified by the DHCP server to renew the IP address lease as long as the access point is running (this parameter is programmed at the DHCP server). For example: Windows 2000 servers typically are set for 3 days.

## Multi-Function LEDs

An model access point has seven LED indicators. Four LEDs exist on the top of the and are visible from wall, ceiling and table-top orientations. Three of these four LEDs are single color activity LEDs, and one is a multi-function red and white status LED. Two LEDs exist on the rear of the access point



and are viewable using a single (customer installed) extended light pipe, adjusted as required to suit above the ceiling installations. An AP3550 model access point houses four LEDs on the bottom/back side of the unit.

## Mesh Networking

Utilize the new mesh networking functionality to allow the access point to function as a bridge to connect two Ethernet networks or as a repeater to extend your network's coverage area without additional cabling. Mesh networking is configurable in two modes. It can be set in a wireless client bridge mode and/or a wireless base bridge mode (which accepts connections from client bridges). These two modes are not mutually exclusive.

In client bridge mode, the access point scans to find other access points using the selected WLAN's ESSID. The access point must go through the association and authentication process to establish a wireless connection. The mesh networking association process is identical to the access point's MU association process. Once the association/authentication process is complete, the wireless client adds the connection as a port on its bridge module. This causes the access point (in client bridge mode) to begin forwarding configuration packets to the base bridge. An access point in base bridge mode allows the access point radio to accept client bridge connections.

The two bridges communicate using the *Spanning Tree Protocol* (STP). The spanning tree determines the path to the root and detects if the current connection is part of a network loop with another connection. Once the spanning tree converges, both access points begin learning which destinations reside on which side of the network. This allows them to forward traffic intelligently.

After the access point (in client bridge mode) establishes at least one wireless connection, it will begin beaconing and accepting wireless connections (if configured to support mobile users). If the access point is configured as both a client bridge and a base bridge, it begins accepting client bridge connections. In this way, the mesh network builds itself over time and distance.

Once the access point (in client bridge mode) establishes at least one wireless connection, it establishes other wireless connections in the background as they become available. In this way, the access point can establish simultaneous redundant links. An access point (in client bridge mode) can establish up to 3 simultaneous wireless connections with other AP3510s or AP3550s. A client bridge always initiates the connections and the base bridge is always the acceptor of the mesh network data proliferating the network.

Since each access point can establish up to 3 simultaneous wireless connections, some of these connections may be redundant. In that case, the STP algorithm determines which links are the redundant links and disables the links from forwarding.

The mesh network using AP35xx is managed by the Extreme Networks Summit WM3000 series controller. For an overview on setting up a mesh network from the Summit WM3000 series controller as well as details on configuring other mesh parameters from the access point CLI, see the "*Summit WM3000 Series Controller System Reference Guide, Software Version 4.0*".

## Additional LAN Subnet

In a typical small office environment (wherein a wireless network is available along with a production WLAN) it is frequently necessary to segment a LAN into two subnets. Consequently, a second LAN is necessary to "segregate" wireless traffic.

The access point has a second LAN subnet enabling administrators to segment the access point's LAN connection into two separate networks. Both LANs can still be active at any given time, but only one can transmit over the access point's physical LAN connection.

## On-board Radius Server Authentication

The access point has the ability to work as a Radius Server to provide user database information and user authentication. Each user is authorized based on the access policies applicable to that user. Access policies allow an administrator to control access to a user groups based on the WLAN configurations.

## Hotspot Support

The access point allows hotspot operators to provide user authentication and accounting without a special client application. Rather than rely on built-in 802.11 security features to control access point association privileges, you can configure a WLAN with no WEP (an open network). The access point issues an IP address to the user using a DHCP server, authenticates the user and grants the user to access the Internet.

If a tourist visits a public hotspot and wants to browse a Web page, they boot their laptop and associate with a local Wi-Fi network by entering a valid SSID. They start a browser, and the hotspot's access controller forces the un-authenticated user to a Welcome page (from the hotspot operator) that allows the user to login with a username and password. In order to send a redirected page (a login page), a TCP termination exists locally on the access point. Once the login page displays, the user enters their credentials. The access point connects to the Radius server and determines the identity of the connected wireless user. Thus, allowing the user to access the Internet once successfully authenticated.

## Dynamic DNS

The access point supports the Dynamic DNS service. *Dynamic DNS* (or DynDNS) is a feature offered by [www.dyndns.com](http://www.dyndns.com) which allows the mapping of domain names to dynamically assigned IP addresses. When the dynamically assigned IP address of a client changes, the new IP address is sent to the DynDNS service and traffic for the specified domain(s) is routed to the new IP address.

## Auto Negotiation

Auto negotiation enables the access point to automatically exchange information (over either its LAN or WAN port) about data transmission speed and duplex capabilities. Auto negotiation is helpful when using the access point in an environment where different devices are connected and disconnected on a regular basis.

## IP Filtering

IP filtering determines which IP packets are processed normally and which are discarded. If discarded, the packet is deleted and completely ignored (as if never received). Optionally apply different criteria to better refine which packets to filter.

IP filtering supports the creation of up to 18 filter rules enforced at layer 3. Once defined (using the access point's SNMP, GUI or CLI), filtering rules can be enforced on the access point's LAN1, LAN2 and

WLAN interfaces. An additional default action is also available denying traffic when the filter rules fail. Lastly, imported and exported configurations retain their defined IP filtering configurations.

## DHCP Lease Information

This release of the access point firmware provides an enhancement to the access point's existing DHCP server functionality, allowing a network administrator to monitor IP address usage. When either (or both) of the access point's LAN interfaces are configured as a DHCP server, a client's IP address lease assignment can now be monitored in respect to its lease period and expiration time. The access point's GUI and CLI interfaces support this feature.

## Configurable MU Idle Timeout

The configurable MU idle timeout allows a MU timeout to be defined separately for individual WLANs. The MU timeout value can be defined using the access point's CLI, GUI and SNMP interfaces. Imported and exported configurations retain their defined MU idle timeout configurations. The default MU idle timeout is 30 minutes for each WLAN.

## Auto Channel Select (ACS) Smart Scan

The access point supports a new *Auto Channel Select* (ACS) feature allowing users to specify an exception list for channel usage. When channel exceptions are defined, the access point skips the channels specified in the list. When the smart scan feature is enabled (it's disabled by default), up to 3 separate channels can be excluded. The exception list is configurable using the access point's CLI, GUI and SNMP interfaces. Imported and exported configurations retain their defined exception list configurations.

## Trusted Host Management

Trusted subnet management restricts AP-51x1 LAN1, LAN2 and WAN interface access (via SNMP, HTTP, HTTPS, Telnet and SSH) to a set of user defined trusted host or subnets. Only hosts with matching subnet (or IP) addresses are able to access the access point. Enabling the feature denies access from any subnet not defined as trusted. Once a set of trusted hosts is defined and applied, the settings can be imported and exported as a part of the access point's configuration import/export functionality.

## Rogue AP Enhancements

The access point now has the option to scan for rogues over all channels on both of the access point's 11a and 11bg radio bands. The switching of radio bands is based on a timer with no user intervention required.

## Radius Time-Based Authentication

An external server maintains a users and groups database used by the access point for access permissions. Various kinds of access policies can be applied to each group. Individual groups can be configured with their own time-based access policy. Each group's policy has a user defined interval

defining the days and hours access is permitted. Authentication requests for users belonging to the group are honored only during these defined hourly intervals.

## QBSS Support

Each access point radio can be configured to optionally allow the access point to communicate channel usage data to associated devices and define the beacon interval used for channel utilization transmissions. The QBSS load represents the percentage of time the channel is in use by the access point and the access point's station count. This information is very helpful in assessing the access point's overall load on a channel, its availability for additional device associations and multi media traffic support.

## Reliable Multicast

This feature enables the AP to reliably transmit multicast transmission to those MUs that have subscribed to them. This is done by converting the multicast packet to unicast packet and then transmitting them. The received multicast packet is then dropped by the AP to prevent the MUs from receiving the transmission twice. Up to a maximum of 16 multicast groups can be supported by this feature. The maximum number of number of simultaneous streams supported is 32.

## Configurable WPA Handshake Retry Levels

The AP has been updated to support configurable WPA handshake retry levels. This is to prevent the MUs from timing out restarting association procedure if it does not receive any of the EAPOL messages from the AP. The retry timeout can be configured to appropriate value between 100ms to 2 seconds so that the EAPOL message is retried by AP before the MU can timeout. Also, the number of retries is configurable in the range of 1 and 10 retries.

## Theory of Operations

To understand access point management and performance alternatives, users need familiarity with its functionality and configuration options. The access point includes features for different interface connections and network management.

The access point uses electromagnetic waves to transmit and receive electric signals without wires. Users communicate with the network by establishing radio links between *mobile units (MUs)* and access points.

The access point uses a digital modulated RF signal to transmit digital data from one device to another. A radio signal begins with a carrier signal that provides the base or center frequency. The digital data signal is encoded onto the carriers using an advanced digital modulation technique as specified in the 802.11a/b/g standards. The radio signal propagates into the air as electromagnetic waves. A receiving antenna (on the MU) in the path of the waves absorbs the waves as electrical signals. The receiving MU interprets (demodulates) the signal by reapplying the direct sequence chipping code. This demodulation results in the original digital data.

The access point uses its environment (the air and certain objects) as the transmission medium. The access point can either transmit in the 2.4 to 2.5-GHz frequency range (802.11b/g radio) or the 5 GHz

frequency range (802.11a radio), the actual range is country-dependent. Extreme Networks devices, like other Ethernet devices, have unique, hardware encoded *Media Access Control (MAC)* or IEEE addresses. MAC addresses determine the device sending or receiving data. A MAC address is a 48-bit number written as six hexadecimal bytes separated by colons. For example: 00:A2:B1:B2:C1:C2.

Also see the following sections:

- [“Cellular Coverage”](#)
- [“MAC Layer Bridging”](#)
- [“Content Filtering”](#)
- [“DHCP Support”](#)
- [“Media Types”](#)
- [“MU Association Process”](#)

## Cellular Coverage

An access point establishes an average communication range with MUs called a *Basic Service Set (BSS)* or cell. When in a particular cell, the MU associates and communicates with the access point supporting the radio coverage area of that cell. Adding access points to a single LAN establishes more cells to extend the range of the network. Configuring the same *ESSID (Extended Service Set Identifier)* on all access point makes them part of the same Wireless LAN.

Access points with the same ESSID define a coverage area. A valid ESSID is an alphanumeric, case-sensitive identifier up to 32 characters. An MU searches for an access point with a matching ESSID and synchronizes (associates) to establish communications. This device association allows MUs within the coverage area to move about or *roam*. As the MU roams from cell to cell, it associates with a different access point. The roam occurs when the MU analyzes the reception quality at a location and determines a different access point provides better signal strength and lower MU load distribution.

If the MU does not find an access point with a workable signal, it can perform a scan to find any AP. As MUs controller APs, the AP updates its association statistics.

The user can configure the ESSID to correspond to up to 16 WLANs on each 802.11a or 802.11b/g radio. A *Wireless Local Area Network (WLAN)* is a data-communications system that flexibly extends the functionalities of a wired LAN. A WLAN does not require lining up devices for line-of-sight transmission, and are thus, desirable. Within the WLAN, roaming users can be handed off from one access point to another like a cellular phone system. WLANs can therefore be configured around the needs of specific groups of users, even when they are not in physical proximity.

## MAC Layer Bridging

The access point provides *MAC layer bridging* between its interfaces. The access point monitors traffic from its interfaces and, based on frame address, forwards the frames to the proper destination. The access point tracks source and destination addresses to provide intelligent bridging as MUs roam or network topologies change. The access point also handles broadcast and multicast messages and responds to MU association requests.

The access point listens to all packets on its LAN and WAN interfaces and builds an address database using MAC addresses. An address in the database includes the interface media that the device uses to associate with the access point. The access point uses the database to forward packets from one interface

to another. The bridge forwards packets addressed to unknown systems to the *Default Interface* (Ethernet).

The access point internal stack interface handles all messages directed to the access point. Each stores information on destinations and their interfaces to facilitate *forwarding*. When a user sends an *ARP* (*Address Resolution Protocol*) request packet, the access point forwards it over all enabled interfaces except over the interface the ARP request packet was received.

On receiving the ARP response packet, the access point database keeps a record of the destination address along with the receiving interface. With this information, the access point forwards any directed packet to the correct destination. Transmitted ARP request packets echo back to other MUs. The access point removes from its database the destination or interface information that is not used for a specified time. The AP refreshes its database when it transmits or receives data from these destinations and interfaces.

## Media Types

The access point radio interface conforms to IEEE 802.11a/b/g specifications. The interface operates at a maximum 54Mbps (802.11a radio) using direct-sequence radio technology. The access point supports multiple-cell operations with fast roaming between cells. Within a direct-sequence system, each cell can operate independently. Adding cells to the network provides an increased coverage area and total system capacity.

The RS-232 serial port provides a *Command Line Interface (CLI)* connection. The serial link supports a direct serial connection (assuming a DB9 connector is used). The access point is a *Data Terminal Equipment (DTE)* device with male pin connectors for the RS-232 port. Connecting the access point to a PC requires a null modem serial cable.

## MU Association Process

An access point recognizes MUs as they begin the association process. An access point keeps a list of the MUs it services. MUs associate with an access point based on the following conditions:

- Signal strength between the access point and MU
- Number of MUs currently associated with the access point
- MUs encryption and authentication capabilities
- MUs supported data rate

MUs perform pre-emptive roaming by intermittently scanning for access point's and associating with the best available access point. Before roaming and associating, MUs perform full or partial scans to collect statistics and determine the direct-sequence channel used by the access point.

Scanning is a periodic process where the MU sends out probe messages on all channels defined by the country code. The statistics enable an MU to reassociate by synchronizing its channel to the access point. The MU continues communicating with that access point until it needs to switch cells or roam.

MUs perform partial scans at programmed intervals, when missing expected beacons or after excessive transmission retries. In a partial scan, the MU scans 's classified as proximate on the access point table. For each channel, the MU tests for *Clear Channel Assessment (CCA)*. The MU broadcasts a probe with the ESSID and broadcast BSS\_ID when the channel is transmission-free. It sends an ACK to a directed probe response from the and updates the table.

An MU can roam within a coverage area by switching access points. Roaming occurs when:

- Unassociated MU attempts to associate or reassociate with an available access point
- Supported rate changes or the MU finds a better transmit rate with another access point
- *RSSI (received signal strength indicator)* of a potential access point exceeds the current access point
- Ratio of good-transmitted packets to attempted-transmitted packets falls below a threshold.

An MU selects the best available access point and adjusts itself to the access point direct-sequence channel to begin association. Once associated, the access point begins forwarding frames addressed to the target MU. Each frame contains fields for the current direct-sequence channel. The MU uses these fields to resynchronize to the access point.





## 3 CLI Reference

The access point *Command Line Interface (CLI)* is accessed through the serial port or a Telnet session. The access point CLI follows the same conventions as the Web-based user interface. The CLI does, however, provide an “escape sequence” to provide diagnostics for problem identification and resolution.

The CLI treats the following as invalid characters:

-> space < > | " & , \ ?

In order to avoid problems when using the CLI, these characters should be avoided.



### NOTE

*AP35xx access points are managed by an Extreme Networks Summit WM3600 or a WM3700 WLAN controller. While the AP can be configured through the AP CLI, the CLI configured parameters on the AP get overwritten by the controller configured parameters if conflicts are found.*

## Connecting to the CLI

### Accessing the CLI through the Serial Port

To connect to the access point CLI through the serial port:

- 1 Connect one end of a null modem serial cable to the access point's serial connector.



### NOTE

*If using an AP3510 model access point, a null modem cable is required. If using an AP3550 model access point, an RJ-45 to Serial cable is required to make the connection.*

- 2 Attach the other end of the null modem serial cable to the serial port of a PC running HyperTerminal or a similar emulation program.
- 3 Set the HyperTerminal program to use 19200 baud, 8 data bits, 1 stop bit, no parity, no flow control, and auto-detect for terminal emulation.
- 4 Press <ESC> or <Enter> to enter into the CLI.
- 5 Enter the default username of admin and the default password of admin123. If this is your first time logging into the access point, you are unable to access any of the access point's commands until the country code is set. A new password will also need to be created.

## Accessing the CLI via Telnet

To connect to the access point CLI through a Telnet connection:

- 1 If this is your first time connecting to your access point, keep in mind the access point's LAN port is set as a DHCP client by default.
- 2 Enter the default username of admin and the default password of admin123. If this is your first time logging into the access point, you are unable to access any of the access point's commands until the country code is set. A new password will also need to be created.

# Admin and Common Commands

## AP35xx>

Displays admin configuration options. The items available under this command are shown below.

### Syntax

|                |  |
|----------------|--|
| <b>help</b>    | Displays general user interface help.    |
| <b>passwd</b>  | Changes the admin password.              |
| <b>summary</b> | Shows a system summary.                  |
| <b>network</b> | Goes to the network submenu              |
| <b>system</b>  | Goes to the system submenu.              |
| <b>stats</b>   | Goes to the stats submenu.               |
| <b>..</b>      | Goes to the parent menu.                 |
| <b>/</b>       | Goes to the root menu.                   |
| <b>save</b>    | Saves the configuration to system flash. |
| <b>quit</b>    | Quits the CLI.                           |

## help

Displays general CLI user interface help.

## Syntax

**help**

## Example

```
admin>help
```

```

?                : display command help - Eg. ?, show ?, s?
* Restriction of "?": : "?" after a function argument is treated
                    : as an argument
                    : Eg. admin<network.lan> set lan enable?
                    : (Here "?" is an invalid extra argument,
                    : because it is after the argument
                    : "enable")

<ctrl-q>         : go backwards in command history
<ctrl-p>         : go forwards in command history

* Note           : 1) commands can be incomplete
                  : - Eg. sh = sho = show
                  : 2) "/" introduces a comment and gets no
                  : response from CLI.
```

```
admin>
```

## **passwd**

Changes the admin password for access point access. This requires typing the old admin password and entering a new password and confirming it. Passwords can be up to 11 characters. The access point CLI treats the following as invalid characters:

-> space < > | " & , \ ?

In order to avoid problems when using the access point CLI, these characters should be avoided.

## **Syntax**

### **passwd**

## **Example**

```
admin>passwd
```

```
Old Admin Password:*****
```

```
New Admin Password (0 - 11 characters):*****
```

```
Verify Admin Password (0 - 11 characters):*****
```

```
Password successfully updated
```

## summary

Displays a summary of high-level characteristics and settings for the WAN, LAN and WLAN.

## Syntax

### summary

## Example

```
admin>summary
```

|                          |              |
|--------------------------|--------------|
| ADP35xx firmware version | 2.3.2.0-015R |
| country code             | us           |
| ap-mode                  | independent  |
| serial number            | 09289-80092  |
| Hw Model                 | AP3510-US    |

|                 |            |
|-----------------|------------|
| WLAN 1:         |            |
| WLAN Name       | WLAN1      |
| ESS ID          | 101        |
| Radio           | 11a, 11b/g |
| VLAN            | VLAN1      |
| Security Policy | Default    |
| QoS Policy      | Default    |

```

LAN1 Name: LAN1
LAN1 Mode: enable
LAN1 IP: 0.0.0.0
LAN1 Mask: 0.0.0.0
LAN1 DHCP Mode: server

```

```

LAN2 Name: LAN2
LAN2 Mode: enable
LAN2 IP: 192.235.1.1
LAN2 Mask: 255.255.255.0
LAN2 DHCP Mode: server

```

| WAN Interface | IP Address   | Network Mask    | Default Gateway | DHCP Client |
|---------------|--------------|-----------------|-----------------|-------------|
| enable        | 172.20.23.10 | 255.255.255.192 | 172.20.23.20    | enable      |

..

This command navigates up one level in the directory structure. This command is available in submenus. It has no effect in the admin menu.

### Example

```
admin(network.lan)>..  
admin(network)>
```

/

This command navigates to the top level in the directory menu. This command is available in submenus. It has no effect in the admin menu.

### Example

```
admin(network.lan)>/  
admin>
```



## **save**

This command saves the current configuration settings. The save command works at all levels of the CLI. The save command must be issued before leaving the CLI for updated settings to be retained.

## **Syntax**

**save**

## **Example**

```
admin>save  
admin>
```

**quit**

Exits the command line interface session and terminates the session. The quit command appears in all of the submenus under admin. In each case, it has the same function, to exit out of the CLI. Once the quit command is executed, the login prompt displays again.

**Example**

```
admin>quit
```

# Network Commands

## **admin>network**

Navigates to the network submenu. The items available under this command are shown below.

|                 |  |
|-----------------|--|
| <b>lan</b>      | Goes to the LAN submenu.                             |
| <b>wan</b>      | Goes to the WAN submenu.                             |
| <b>wireless</b> | Goes to the Wireless Configuration submenu.          |
| <b>firewall</b> | Goes to the firewall submenu.                        |
| <b>router</b>   | Goes to the router submenu.                          |
| <b>ipfilter</b> | Goes to the IP Filtering submenu.                    |
| <b>..</b>       | Goes to the parent menu.                             |
| <b>/</b>        | Goes to the root menu.                               |
| <b>save</b>     | Saves the current configuration to the system flash. |
| <b>quit</b>     | Quits the CLI and exits the current session.         |

## Network LAN Commands

### **admin(network.lan)>**

Navigates to the LAN submenu. The items available under this menu are shown below.

|                     |  |
|---------------------|--|
| <b>show</b>         | Shows current access point LAN parameters. |
| <b>set</b>          | Sets LAN parameters.                       |
| <b>bridge</b>       | Goes to the mesh configuration submenu.    |
| <b>wlan-mapping</b> | Goes to the WLAN/Lan/Vlan Mapping submenu. |
| <b>dhcp</b>         | Goes to the LAN DHCP submenu.              |
| <b>type-filter</b>  | Goes to the Ethernet Type Filter submenu.  |
| <b>..</b>           | Goes to the parent menu.                   |
| <b>/</b>            | Goes to the root menu.                     |
| <b>save</b>         | Saves the configuration to system flash.   |
| <b>quit</b>         | Quits the CLI.                             |

## **admin(network.lan)> show**

Displays the access point LAN settings.

### **Syntax**

**show**     Shows the settings for the access point LAN1 and LAN2 interfaces.

### **Example**

```
admin(network.lan)>show
```

```
LAN On Ethernet Port           : LAN1
LAN Ethernet Timeout           : disable

802.1x Port Authentication:
    Username                    : admin
    Password                    : *****

Auto-negotiation               : disable
Speed                          : 100M
Duplex                          : full

** LAN1 Information **
LAN Name                       : LAN1
LAN Interface                   : enable
802.11q Trunking                : disable

LAN IP mode                     : DHCP client
IP Address                     : 192.168.0.1
Network Mask                    : 255.255.255.255
Default Gateway                 : 192.168.0.1
Domain Name                     :
Primary DNS Server              : 192.168.0.1
Secondary DNS Server            : 192.168.0.2
WINS Server                     : 192.168.0.254

** LAN2 Information **
LAN Name                       : LAN2
LAN Interface                   : disable
802.11q Trunking                : disable

LAN IP mode                     : DHCP server
IP Address                     : 192.168.1.1
Network Mask                    : 255.255.255.255
Default Gateway                 : 192.168.1.1
Domain Name                     :
Primary DNS Server              : 192.168.0.2
Secondary DNS Server            : 192.168.0.3
WINS Server                     : 192.168.0.255

admin(network.lan)>
```

**admin(network.lan)> set**

Sets the LAN parameters for the LAN port.

**Syntax**

|            |                          |             |  |
|------------|--------------------------|-------------|--|
| <b>set</b> | <b>lan</b>               | <mode>      | Enables or disables the access point LAN interface.  |
|            | <b>name</b>              | <idx-name > | Defines the LAN name by index.   |
|            | <b>ethernet-port-lan</b> | <idx>       | Defines which LAN (LAN1 or LAN2) is active on the Ethernet port.   |
|            | <b>timeout</b>           | <seconds>   | Sets the interval (in seconds) the access point uses to terminate its LAN interface if no activity is detected for the specified interval. |
|            | <b>trunking</b>          | <mode>      | Enables or disables 802.11q Trunking over the access point LAN port.   |
|            | <b>auto-negotiation</b>  | <mode>      | Enables or disables auto-negotiation for the access point LAN port.  |
|            | <b>ipfpolicy</b>         | <name>      | Sets the IP-Filtering Policy name.   |
|            | <b>speed</b>             | <mbps>      | Defines the access point LAN port speed as either 10 Mbps or 100 Mbps.   |
|            | <b>duplex</b>            | <mode>      | Defines the access point LAN port duplex as either half or full.   |
|            | <b>username</b>          | <name>      | Specifies the user name for 802.1x port authentication over the LAN interface.   |
|            | <b>passwd</b>            | <password>  | The 0-32 character password for the username for the 802.1x port.  |
|            | <b>ip-mode</b>           | <ip>        | Defines the access point LAN port IP mode.   |
|            | <b>ipadr</b>             | <ip>        | Sets the IP address used by the LAN port.  |
|            | <b>mask</b>              | <ip>        | Defines the IP address used for access point LAN port network mask.  |
|            | <b>dgw</b>               | <ip>        | Sets the Gateway IP address used by the LAN port.  |
|            | <b>domain</b>            | <name>      | Specifies the domain name used by the access point LAN port.   |
|            | <b>dns</b>               | <ip>        | Defines the IP address of the primary and secondary DNS servers used by the LAN port.  |
|            | <b>wins</b>              | <ip>        | Defines the IP address of the WINS server used by the LAN port.  |

**Example**

```
admin(network.lan)>
```

```
admin(network.lan)>set lan 1 enable
admin(network.lan)>set name 1 engineering
admin(network.lan)>set ethernet-port-lan 1
admin(network.lan)>set timeout 45
admin(network.lan)>set trunking 1 disable
admin(network.lan)>set auto-negotiation disable
admin(network.lan)>set speed 100M
admin(network.lan)>set duplex full
admin(network.lan)>set dns 1 192.168.0.1
admin(network.lan)>set dns 2 192.168.0.2
admin(network.lan)>set wins 1 192.168.0.254
admin(network.lan)>set trunking disable
admin(network.lan)>set username phil
admin(network.lan)>set passwd ea0258c1
```

Related Commands:

**show** Shows the current settings for the access point LAN port.

## Network LAN, Bridge Commands

### **admin(network.lan.bridge)>**

Displays the access point Bridge submenu.

|             |   |
|-------------|---|
| <b>show</b> | Displays the mesh configuration parameters for the access point's LANs. |
| <b>set</b>  | Sets the mesh configuration parameters for the access point's LANs.     |
| <b>..</b>   | Moves to the parent menu.   |
| <b>/</b>    | Goes to the root menu.  |
| <b>save</b> | Saves the configuration to system flash.                                |
| <b>quit</b> | Quits the CLI and exits the session.                                    |

**admin(network.lan.bridge)> show**

Displays mesh bridge configuration parameters for the access point's LANs.

**Syntax**

**show**            Displays mesh bridge configuration parameters for the access point's LANs.

**Example**

```
admin(network.lan.bridge)>show
```

```
** LAN1 Bridge Configuration **
Bridge Priority           :32768
Hello Time (seconds)     :2
Message Age Time (seconds) :20
Forward Delay Time (seconds) :15

Entry Ageout Time (seconds) :300

** LAN2 Bridge Configuration **
Bridge Priority           :32768
Hello Time (seconds)     :2
Message Age Time (seconds) :20
Forward Delay Time (seconds) :15

Entry Ageout Time (seconds) :300
```



## admin(network.lan.bridge)> set

Sets the mesh configuration parameters for the access point's LANs.

### Syntax

|            |                 |           |           |   |
|------------|-----------------|-----------|-----------|---|
| <b>set</b> | <b>priority</b> | <LAN-idx> | <seconds> | Sets bridge priority time in seconds (0-65535) for specified LAN.           |
|            | <b>hello</b>    | <LAN-idx> | <seconds> | Sets bridge hello time in seconds (0-10) for specified LAN.                 |
|            | <b>msgage</b>   | <LAN-idx> | <seconds> | Sets bridge message age time in seconds (6-40) for specified LAN.           |
|            | <b>fwddelay</b> | <LAN-idx> | <seconds> | Sets bridge forward delay time in seconds (4-30) for specified LAN.         |
|            | <b>ageout</b>   | <LAN-idx> | <seconds> | Sets bridge forward table entry time in seconds (4-3600) for specified LAN. |

### Example

```
admin(network.lan.bridge)>set priority 2 32768
admin(network.lan.bridge)>set hello 2 2
admin(network.lan.bridge)>set msgage 2 20
admin(network.lan.bridge)>set fwddelay 2 15
admin(network.lan.bridge)>set ageout 2 300
```

```
admin(network.lan.bridge)>show
```

```
** LAN1 Mesh Configuration **
```

```
Bridge Priority           :32768
Hello Time (seconds)     :2
Message Age Time (seconds) :20
Forward Delay Time (seconds) :15
```

```
Entry Ageout Time (seconds) :300
```

```
** LAN2 Mesh Configuration **
```

```
Bridge Priority           :32768
Hello Time (seconds)     :2
Message Age Time (seconds) :20
Forward Delay Time (seconds) :15
```

```
Entry Ageout Time (seconds) :300
```

## Network LAN, WLAN-Mapping Commands

### **admin(network.lan)>wlan-mapping**

Navigates to the WLAN/Lan/Vlan Mapping submenu.

|                 |  |
|-----------------|--|
| <b>show</b>     | Displays the VLAN list currently defined for the access point. |
| <b>set</b>      | Sets the access point VLAN configuration.                      |
| <b>create</b>   | Creates a new access point VLAN.                               |
| <b>edit</b>     | Edits the properties of an existing access point VLAN.         |
| <b>delete</b>   | Deletes a VLAN.  |
| <b>lan-map</b>  | Maps access point existing WLANs to an enabled LAN.            |
| <b>vlan-map</b> | Maps access point existing WLANs to VLANs.                     |
| <b>..</b>       | Moves to the parent menu.                                      |
| <b>/</b>        | Goes to the root menu.   |
| <b>save</b>     | Saves the configuration to system flash.                       |
| <b>quit</b>     | Quits the CLI and exits the session.                           |

## admin(network.lan.wlan-mapping)> show

Displays the VLAN list currently defined for the access point.. These parameters are defined with the set command.

### Syntax

|             |                 |   |
|-------------|-----------------|---|
| <b>show</b> | <b>name</b>     | Displays the existing list of VLAN names.       |
|             | <b>vlan-cfg</b> | Shows WLAN-VLAN mapping and VLAN configuration. |
|             | <b>lan-wlan</b> | Displays a WLAN-LAN mapping summary.            |
|             | <b>wlan</b>     | Displays the WLAN summary list.                 |

### Example

```
admin(network.lan.wlan-mapping)>show name
```

| Index | VLAN ID | VLAN Name |
|-------|---------|-----------|
| 1     | 1       | VLAN_1    |
| 2     | 2       | VLAN_2    |
| 3     | 3       | VLAN_3    |
| 4     | 4       | VLAN_4    |

```
admin(network.lan.wlan-mapping)>show vlan-cfg
```

```
Management VLAN Tag      :1
Native VLAN Tag          :2
WLAN                     :WLAN1
mapped to VLAN           :VLAN 2
VLAN Mode                :static
```

```
admin(network.lan.wlan-mapping)>show lan-wlan
```

```
WLANs on LAN1:
      :WLAN1
      :WLAN2
      :WLAN3
```

```
WLANs on LAN2:
```

```
admin(network.lan.wlan-mapping)>show wlan
```

```
WLAN1:
WLAN Name      :WLAN1
ESSID          :101
Radio          :
VLAN           :
Security Policy :Default
QoS Policy     :Default
```

**admin(network.lan.wlan-mapping)> set**

Sets VLAN parameters for the access point.

**Syntax**

|            |                   |            |  |
|------------|-------------------|------------|--|
| <b>set</b> | <b>mgmt- tag</b>  | <id>       | Defines the Management VLAN tag (1-4095).                    |
|            | <b>native-tag</b> | <id>       | Sets the Native VLAN tag (1-4095).                           |
|            | <b>mode</b>       | <wlan-idx> | Sets WLAN VLAN mode (WLAN 1-16) to either dynamic or static. |

**Example**

```
admin(network.lan.wlan-mapping)>set mgmt-tag 1
admin(network.lan.wlan-mapping)>set native-tag 2
admin(network.lan.wlan-mapping)>set mode 1 static
```

```
admin(network.lan.wlan-mapping)>show vlan-cfg
```

```
Management VLAN Tag      :1
Native VLAN Tag          :2
WLAN                     :WLAN1
mapped to VLAN           :VLAN 2
VLAN Mode                :static
```

## **admin(network.lan.wlan-mapping)> create**

Creates a VLAN for the access point.

### **Syntax**

|               |                  |        |   |
|---------------|------------------|--------|---|
| <b>create</b> | <b>vlan-id</b>   | <id>   | Defines the VLAN ID (1-4095).                               |
|               | <b>vlan-name</b> | <name> | Specifies the name of the VLAN (1-31 characters in length). |

### **Example**

```
admin(network.lan.wlan-mapping)>  
admin(network.lan.wlan-mapping)>create 5 vlan-5
```

**admin(network.lan.wlan-mapping)> edit**

Modifies a VLAN's name and ID.

**Syntax**

**edit**      **name**      <vlan-idx> <name>    Modifies an existing VLAN name (1-31 characters in length).  
              **id**        <vlan-idx> <vlan-id>   Modifies an existing VLAN ID (1-4095) characters in length).

**Example**

```
admin(network.lan.wlan-mapping)>show name
```

```
-----
Index   VLAN ID   VLAN Name
-----
1       1         Vlan_001
2       2         Vlan_002
3       3         Vlan_003
```

```
admin(network.lan.wlan-mapping)>edit name 1 VlanConfRoom
```

```
admin(network.lan.wlan-mapping)>show name
```

```
-----
Index   VLAN ID   VLAN Name
-----
1       1         VlanConfRoom
2       2         Vlan_002
3       3         Vlan_003
```

## **admin(network.lan.wlan-mapping)> delete**

Deletes a specific VLAN or all VLANs.

### **Syntax**

**delete**     < VLAN id> Deletes a specific VLAN ID (1-16).  
**all**         Deletes all defined VLANs.

### **Example**

```
admin(network.lan.wlan-mapping)>show name
```

| Index | VLAN ID | VLAN Name    |
|-------|---------|--------------|
| 1     | 1       | VlanConfRoom |
| 2     | 2       | Vlan_002     |
| 3     | 3       | Vlan_003     |

```
admin(network.lan.wlan-mapping)>delete 2
admin(network.lan.wlan-mapping)>show name
```

| Index | VLAN ID | VLAN Name    |
|-------|---------|--------------|
| 1     | 1       | VlanConfRoom |
| 3     | 3       | Vlan_003     |

**admin(network.lan.wlan-mapping)> lan-map**

Maps an access point VLAN to a WLAN.

**Syntax**

|                |                            |  |
|----------------|----------------------------|--|
| <b>lan-map</b> | <wlan-name> <lan-name><cr> | Maps an existing WLAN to an enabled LAN. All names and IDs are case-sensitive. |
|                | <wlan-name>                | Displays existing WLAN name.   |
|                | <lan-name>                 | Defines enabled LAN name. All names and IDs are case-sensitive.                |

**Example**

```
admin(network.lan.wlan-mapping)>lan-map wlan1 lan1
```



## **admin(network.lan.wlan-mapping)> vlan-map**

Maps an access point VLAN to a WLAN.

### **Syntax**

|   |   |
|---|---|
| <b>vlan-map</b> <wlan-name> <vlan-name><cr> | Maps an existing WLAN to an enabled VLAN. All names and IDs are case-sensitive. |
| <wlan-name>                                 | Displays existing WLAN name.  |
| <vlan-name>                                 | Maps an existing WLAN to an enabled VLAN. All names and IDs are case-sensitive. |

### **Example**

```
admin(network.lan.wlan-mapping)>vlan-map wlan1 vlan1
```

## Network LAN, DHCP Commands

### **admin(network.lan)> dhcp**

Navigates to the access point DHCP submenu. The items available are displayed below.

|               |  |
|---------------|--|
| <b>show</b>   | Displays DHCP parameters.                |
| <b>set</b>    | Sets DHCP parameters.                    |
| <b>add</b>    | Adds static DHCP address assignments.    |
| <b>delete</b> | Deletes static DHCP address assignments. |
| <b>list</b>   | Lists static DHCP address assignments.   |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI and exits the session.     |

## **admin(network.lan.dhcp)> show**

Displays DHCP parameter settings for the access point. These parameters are defined with the set command.

### **Syntax**

**show**                Displays DHCP parameter settings for the access point. These parameters are defined with the set command.

### **Example**

```
admin(network.lan.dhcp)>show
**LAN1 DHCP Information**
DHCP Address Assignment Range:
    Starting IP Address   : 192.168.0.100
    Ending IP Address     : 192.168.0.254

Lease Time               : 86400

**LAN2 DHCP Information**
DHCP Address Assignment Range:
    Starting IP Address   : 192.168.0.100
    Ending IP Address     : 192.168.0.254

Lease Time               : 86400
```

**admin(network.lan.dhcp)> set**

Sets DHCP parameters for the LAN port.

**Syntax**

|                  |           |          |       |   |
|------------------|-----------|----------|-------|---|
| <b>set range</b> | <LAN-idx> | <ip1>    | <ip2> | Sets the DHCP assignment range from IP address <ip1> to IP address <ip2> for the specified LAN.   |
| <b>lease</b>     | <LAN-idx> | <period> |       | Sets the DHCP lease time <period> in seconds (120-999999) for the specified LAN (1-LAN1, 2-LAN2). |

**Example**

```
admin(network.lan.dhcp)>set range 1 192.168.0.100 192.168.0.254
admin(network.lan.dhcp)>set lease 1 86400
```

```
admin(network.lan.dhcp)>show
**LAN1 DHCP Information**
DHCP Address Assignment Range:
    Starting IP Address   : 192.168.0.100
    Ending IP Address     : 192.168.0.254

Lease Time                : 86400
```

## **admin(network.lan.dhcp)> add**

Adds static DHCP address assignments.

### **Syntax**

**add** <LAN-idx> <mac> <ip> Adds a reserved static IP address to a MAC address for the specified LAN.

### **Example**

```
admin(network.lan.dhcp)>add 1 00A0F8112233 192.160.24.6
admin(network.lan.dhcp)>add 1 00A0F1112234 192.169.24.7
admin(network.lan.dhcp)>list 1
```

```
-----
Index   MAC Address      IP Address
-----
1       00A0F8112233     192.160.24.6
2       00A0F8112234     192.169.24.7
```

**admin(network.lan.dhcp)> delete**

Deletes static DHCP address assignments.

**Syntax**

**delete**    <LAN-idx> <idx>            Deletes the static DHCP address entry for the specified LAN (1-LAN1, 2-LAN2) and DHCP entry index (1-30).  
              <LAN-idx> all                Deletes all static DHCP addresses.

**Example**

```
admin(network.lan.dhcp)>list 1
```

```
-----
Index   MAC Address      IP Address
-----
1       00A0F8112233     10.1.2.4
2       00A0F8102030     10.10.1.2
3       00A0F8112234     10.1.2.3
4       00A0F8112235     192.160.24.6
5       00A0F8112236     192.169.24.7
```

```
admin(network.lan.dhcp)>delete 1
```

```
-----
index   mac address      ip address
-----
1       00A0F8102030     10.10.1.2
2       00A0F8112234     10.1.2.3
3       00A0F8112235     192.160.24.6
4       00A0F8112236     192.169.24.7
```

```
admin(network.lan.dhcp)>delete 1 all
```

```
-----
index   mac address      ip address
-----
```

## **admin(network.lan.dhcp)> list**

Lists static DHCP address assignments.

### **Syntax**

**list**            <LAN-idx> <cr>      Lists the static DHCP address assignments for the specified LAN (1-LAN1, 2 LAN2).

### **Example**

```
admin(network.lan.dhcp)>list 1
```

| ----- |              |              |
|-------|--------------|--------------|
| Index | MAC Address  | IP Address   |
| ----- |              |              |
| 1     | 00A0F8112233 | 10.1.2.4     |
| 2     | 00A0F8102030 | 10.10.1.2    |
| 3     | 00A0F8112234 | 10.1.2.3     |
| 4     | 00A0F8112235 | 192.160.24.6 |
| 5     | 00A0F8112236 | 192.169.24.7 |

```
admin(network.lan.dhcp)>
```

## Network Type Filter Commands

### **admin(network.lan)> type-filter**

Navigates to the access point Type Filter submenu. The items available under this command include:

|               |  |
|---------------|--|
| <b>show</b>   | Displays the current Ethernet Type exception list. |
| <b>set</b>    | Defines Ethernet Type Filter parameters.           |
| <b>add</b>    | Adds an Ethernet Type Filter entry.                |
| <b>delete</b> | Removes an Ethernet Type Filter entry.             |
| <b>..</b>     | Goes to the parent menu.                           |
| <b>/</b>      | Goes to the root menu.                             |
| <b>save</b>   | Saves the configuration to system flash.           |
| <b>quit</b>   | Quits the CLI.                                     |



## **admin(network.lan.type-filter)> show**

Displays the access point's current Ethernet Type Filter configuration.

### **Syntax**

**show** <LAN-idx> Displays the existing Type-Filter configuration for the specified LAN.

### **Example**

```
admin(network.lan.type-filter)>show 1
```

```
Ethernet Type Filter mode           : allow
```

```
-----  
index          ethernet type  
-----
```

```
1              8137
```

**admin(network.lan.type-filter)> set**

Allows or denies the access point from processing a specified Ethernet data type for the specified LAN.

**Syntax**

```
set mode          <LAN-idx>  <filter mode>
                        allow/deny
```

**Example**

```
admin(network.lan.type-filter)>set mode 1 allow
```

## **admin(network.lan.type-filter)> add**

Adds an Ethernet Type Filter entry.

### **Syntax**

**add** <LAN-idx>            <type>            Adds entered Ethernet Type to list of data types either allowed or denied access point processing permissions for the specified LAN (either LAN1 or LAN2).

### **Example**

```
admin(network.lan.type-filter)>
```

```
admin(network.wireless.type-filter)>add 1 8137
```

```
admin(network.wireless.type-filter)>add 2 0806
```

```
admin(network.wireless.type-filter)>show 1
```

```
Ethernet Type Filter mode                               : allow
```

```
-----  
index                               ethernet type  
-----
```

```
1                                   8137
```

```
2                                   0806
```

```
3                                   0800
```

```
4                                   8782
```

**admin(network.lan.type-filter)> delete**

Removes an Ethernet Type Filter entry individually or the entire Type Filter list.

**Syntax**

|               |           |             |   |
|---------------|-----------|-------------|---|
| <b>delete</b> | <LAN-idx> | <entry-idx> | Deletes the specified Ethernet Type entry index (1 through 16). |
|               | <LAN-idx> | all         | Deletes all Ethernet entries currently in list.                 |

**Example**

```
admin(network.lan.type-filter)>delete 1 1
admin(network.lan.type-filter)>show 1
```

```
Ethernet Type Filter mode          : allow
```

```
-----
index          ethernet type
-----
1              0806
2              0800
3              8782
```

```
admin(network.lan.type-filter)>delete 2 all
admin(network.lan.type-filter)>show 2
```

```
Ethernet Type Filter mode          : allow
```

```
-----
index          ethernet type
-----
```

## Network WAN Commands

### **admin(network)> wan**

Navigates to the WAN submenu. The items available under this command are shown below.

|                |   |
|----------------|---|
| <b>show</b>    | Displays the access point WAN configuration and the access point's current PPPoE configuration. |
| <b>set</b>     | Defines the access point's WAN and PPPoE configuration.   |
| <b>nat</b>     | Displays the NAT submenu, wherein Network Address Translations (NAT) can be defined.            |
| <b>vpn</b>     | Goes to the VPN submenu, where the access point VPN tunnel configuration can be set.            |
| <b>content</b> | Goes to the outbound content filtering menu.  |
| <b>dyndns</b>  | Displays the Dynamic DNS submenu, wherein dyndns settings can be defined.                       |
| <b>..</b>      | Goes to the parent menu.  |
| <b>/</b>       | Goes to the root menu.  |
| <b>save</b>    | Saves the current configuration to the access point system flash.                               |
| <b>quit</b>    | Quits the CLI and exits the current session.  |

**admin(network.wan)> show**

Displays the access point WAN port parameters.

**Syntax**

**show** Shows the general IP parameters for the WAN port along with settings for the WAN interface.

**Example**

```
admin(network.wan)>show
```

```
Status                                     : enable
WAN DHCP Client Mode                     : enable
IP Address                               : 157.235.112.32
Network Mask                             : 0.0.0.0
Default Gateway                           : 0.0.0.0
Primary DNS Server                        : 0.0.0.0
Secondary DNS Server                      : 0.0.0.0

Auto-negotiation                         : disable
Speed                                    : 100M
Duplex                                    : full

WAN IP 2                                 : disable
WAN IP 3                                 : disable
WAN IP 4                                 : disable
WAN IP 5                                 : disable
WAN IP 6                                 : disable
WAN IP 7                                 : disable
WAN IP 8                                 : disable

PPPoE Mode                               : enable
PPPoE User Name                           : JohnDoe
PPPoE Password                           : *****
PPPoE keepalive mode                      : enable
PPPoE Idle Time                           : 600
PPPoE Authentication Type                 : chap
PPPoE State
```

```
admin(network.wan)>
```

## admin(network.wan)> set

Defines the configuration of the access point WAN port.

### Syntax

|            |                         |                |                |   |
|------------|-------------------------|----------------|----------------|---|
| <b>set</b> | <b>wan</b>              | enable/disable |                | Enables or disables the access point WAN port.  |
|            | <b>dhcp</b>             | enable/disable |                | Enables or disables WAN DHCP Client mode.   |
|            | <b>ipadr</b>            | <idx>          | <a.b.c.d>      | Sets up to 8 (using <idx> from 1 to 8) IP addresses <a.b.c.d> for the access point WAN interface.   |
|            | <b>mask</b>             | <a.b.c.d>      |                | Sets the subnet mask for the access point WAN interface.  |
|            | <b>dgw</b>              | <a.b.c.d>      |                | Sets the default gateway IP address to <a.b.c.d>.   |
|            | <b>dns</b>              | <idx>          | <a.b.c.d>      | Sets the IP address of one or two DNS servers, where <idx> indicates either the primary (1) or secondary (2) server, and <a.b.c.d> is the IP address of the server. |
|            | <b>auto-negotiation</b> | enable/disable |                | Enables or disables auto-negotiation for the access point WAN port.   |
|            | <b>speed</b>            | <mbps>         |                | Defines the access point WAN port speed as either 10 Mbps or 100 Mbps.  |
|            | <b>duplex</b>           | <mode>         |                | Defines the access point WAN port duplex as either half or full.  |
|            | <b>pppoe</b>            | mode           | enable/disable | Enables or disables PPPoE.  |
|            |                         | user           | <name>         | Sets PPPoE user name.   |
|            |                         | passwd         | <password>     | Defines the PPPoE password.   |
|            |                         | ka             | enable/disable | Enables or disables PPPoE keepalive.  |
|            |                         | idle           | <time>         | Sets PPPoE idle time.   |
|            |                         | type           | <auth-type>    | Sets PPPoE authentication type.   |

### Example

```
admin(network.wan)>
```

```
admin(network.wan)>set dhcp disable
admin(network.wan)>set ipadr 157.169.22.5
admin(network.wan)>set dgw 157.169.22.1
admin(network.wan)>set dns 1 157.169.22.2
admin(network.wan)>set auto-negotiation disable
admin(network.wan)>set speed 10M
admin(network.wan)>set duplex half
admin(network.wan)>set mask 255.255.255.000
admin(network.wan)>set pppoe mode enable
admin(network.wan)>set pppoe type chap
admin(network.wan)>set pppoe user jk
admin(network.wan)>set pppoe passwd @$goodpassword%$#
admin(network.wan)>set pppoe ka enable
admin(network.wan)>set pppoe idle 600
```

## Network WAN NAT Commands

### **admin(network.wan)> nat**

Navigates to the NAT submenu. The items available under this command are shown below.

|               |   |
|---------------|---|
| <b>show</b>   | Displays the access point's current NAT parameters for the specified index. |
| <b>set</b>    | Defines the access point NAT settings.                                      |
| <b>add</b>    | Adds NAT entries.   |
| <b>delete</b> | Deletes NAT entries.  |
| <b>list</b>   | Lists NAT entries.  |
| <b>..</b>     | Goes to the parent menu.  |
| <b>/</b>      | Goes to the root menu.  |
| <b>save</b>   | Saves the configuration to system flash.                                    |
| <b>quit</b>   | Quits the CLI.  |



## **admin(network.wan.nat)> show**

Displays access point NAT parameters for the specified NAT index.

### **Syntax**

**show** <idx> <cr> Displays access point NAT parameters for the specified NAT index.

### **Example**

```
admin(network.wan.nat)>show 2
```

```
WAN IP Mode           : enable
WAN IP Address        : 157.235.91.2
NAT Type              : 1-to-many
Inbound Mappings      : Port Forwarding
```

```
unspecified port forwarding mode : enable
unspecified port fwd. ip address : 111.223.222.1
one to many nat mapping
```

```
-----
LAN No.           WAN IP
-----
1                 157.235.91.2
2                 157.235.91.2
```

```
admin(network.wan.nat)>
```

**admin(network.wan.nat)> set**

Sets NAT inbound and outbound parameters.

**Syntax**

**set**      <type>          <ip>              <inb>              <outb>  
 Sets the type of NAT translation for WAN address index <idx> (1-8) to <type> (none, 1-to-1, or 1-to-many).  
 Sets the NAT IP mapping associated with WAN address <idx> to the specified IP address <ip>. Sets the  
 inbound IP address for specified index <index> <ip address>. Sets the inbound mode for specified index  
 <index> <enable/disable>. Sets the outbound IP address for specified index <index> <ip address>. Sets the  
 outbound NAT destination <LAN1 or LAN2> <WAN ip 1-8 or None>.

**Example**

```
admin(network.wan.nat)>set type 2 1-to-many
admin(network.wan.nat)>set ip 2 10.1.1.1

admin(network.wan.nat)>show 2

WAN IP Mode                : enable
WAN IP Address              : 157.235.91.2
NAT Type                    : 1-to-many
Inbound Mappings            : Port Forwarding

unspecified port forwarding mode : enable
unspecified port fwd. ip address : 111.223.222.1
one to many nat mapping
```

```
-----
LAN No.           WAN IP
-----
1                 157.235.91.2
2                 10.1.1.1
```

## **admin(network.wan.nat)> add**

Adds NAT entries.

### **Syntax**

**add**     <idx>        <name>        <tran>        <port1>        <port2>        <ip>        <dst\_port>  
Sets the WAN index <idx> (1-8). Defines the <name> of the WAN NAT list (1-7). Sets the transportation protocol <tran> (tcp, udp, icmp, ah, esp, gre or all). Sets the starting port <port1> number and ending port number <port2> in the available port range (1-65535). Sets the internal IP address <ip>. Sets the internal translation port <dst\_port> (1-65535).

### **Example**

```
admin(network.wan.nat)>add 1 indoors udp 20 29 10.10.2.2
```

```
admin(network.wan.nat)>list 1
```

```
-----  
index   name    prot   start port   end port   internal ip   translation port  
-----  
1       indoor  udp    20           29         10.10.2.2    0
```

Related Commands:

**delete**            Deletes one of the inbound NAT entries from the list.  
**list**              Displays the list of inbound NAT entries.

**admin(network.wan.nat)> delete**

Deletes NAT entries.

**Syntax**

**delete**    <idx>    <entry>    Deletes a specified NAT index entry <entry> associated with the WAN.  
              <idx>    all            Deletes all NAT entries associated with the WAN.

**Example**

```
admin(network.wan.nat)>list 1
```

```
-----
index   name      prot   start port   end port   internal ip   translation port
-----
1       special tcp    20           21          192.168.42.16  21
```

```
admin(network.wan.nat)>delete 1 1
```

```
admin(network.wan.nat)>list 1
```

```
-----
index   name      prot   start port   end port   internal ip   translation port
-----
```

Related Commands:

**add**            Adds entries to the list of inbound NAT entries.  
**list**           Displays the list of inbound NAT entries.

## **admin(network.wan.nat)> list**

Lists access point NAT entries for the specified index.

### **Syntax**

**list**        <idx>    Lists the inbound NAT entries associated with the WAN index (1-8).

**delete**        Deletes inbound NAT entries from the list.

**add**            Adds entries to the list of inbound NAT entries.

### **Example**

```
admin(network.wan.nat)>list 1
```

```
-----  
index   name      transport  start port  end port   internal ip  translation port  
-----  
1       special tcp      20         21         192.168.42.16  21
```

## Network WAN, VPN Commands

### **admin(network.wan)> vpn**

Navigates to the VPN submenu. The items available under this command include:

|                 |  |
|-----------------|--|
| <b>add</b>      | Adds VPN tunnel entries.                               |
| <b>set</b>      | Sets key exchange parameters.                          |
| <b>delete</b>   | Deletes VPN tunnel entries.                            |
| <b>list</b>     | Lists VPN tunnel entries                               |
| <b>reset</b>    | Resets all VPN tunnels.                                |
| <b>stats</b>    | Lists security association status for the VPN tunnels. |
| <b>ikestate</b> | Displays an Internet Key Exchange (IKE) summary.       |
| <b>..</b>       | Goes to the parent menu.                               |
| <b>/</b>        | Goes to the root menu.                                 |
| <b>save</b>     | Saves the configuration to system flash.               |
| <b>quit</b>     | Quits the CLI.   |

## **admin(network.wan.vpn)> add**

Adds a VPN tunnel entry.

### **Syntax**

**add** <name> <subnet-idx> <local WAN IP> <remote subnet> <remote subnet mask> <remote gateway>

Creates a tunnel <name> (1 to 13 characters) to gain access through local WAN IP <local WAN IP> from the remote subnet with IP address <remote subnet> and subnet mask <remote subnet mask> using the remote gateway <remote gateway>.

### **Example**

```
admin(network.wan.vpn)>add 2 SJSharkey 209.235.44.31 206.107.22.46 255.255.255.224  
206.107.22.1
```

If tunnel type is Manual, proper SPI values and Keys must be configured after adding the tunnel

```
admin(network.wan.vpn)>
```

**admin(network.wan.vpn)> set**

Sets VPN entry parameters.

**Syntax**

|            |              |           |                      |   |
|------------|--------------|-----------|----------------------|---|
| <b>set</b> | type         | <name>    | <tunnel type>        | Sets the tunnel type <name> to <i>Auto</i> or <i>Manual</i> for the specified tunnel name.  |
|            | authalgo     | <name>    | <authalgo>           | Sets the authentication algorithm for <name> to ( <i>None</i> , <i>MD5</i> , or <i>SHA1</i> ).  |
|            | authkey      | <name>    | <dir> <authkey>      | Sets the AH authentication key (if type is <i>Manual</i> ) for tunnel <name> with the direction set to <i>IN</i> or <i>OUT</i> , and the manual authentication key set to <authkey>. (The key size is 32 hex characters for MD5, and 40 hex characters for SHA1).   |
|            | esp-type     | <name>    | <esptype>            | Sets the Encapsulating Security Payload (ESP) type. Options include <i>None</i> , <i>ESP</i> , or <i>ESP-AUTH</i> .   |
|            | esp-encalgo  | <name>    | <escalgo>            | Sets the ESP encryption algorithm. Options include <i>DES</i> , <i>3DES</i> , <i>AES128</i> , <i>AES192</i> , or <i>AES256</i> .  |
|            | esp-enckey   | <name>    | <dir> <enckey>       | Sets the Manual Encryption Key in ASCII for tunnel <name> and direction <i>IN</i> or <i>OUT</i> to the key <enc-key>. The size of the key depends on the encryption algorithm.<br><ul style="list-style-type: none"> <li>- 16 hex characters for DES</li> <li>- 48 hex characters for 3DES</li> <li>- 32 hex characters for AES128</li> <li>- 48 hex characters for AES192</li> <li>- 64 hex characters for AES256</li> </ul> |
|            | esp-authalgo | <name>    | <authalgo>           | Sets the ESP authentication algorithm. Options include <i>MD5</i> or <i>SHA1</i> .  |
|            | esp-authkey  | <name>    | <dir> <authkey>      | Sets ESP Authentication key <name> either for <i>IN</i> or <i>OUT</i> direction to <auth-key>, an ASCII string of hex characters. If authalgo is set to <i>MD5</i> , then provide 32 hex characters. If authalgo is set to <i>SHA1</i> , provide 40 hex characters.   |
|            | spi          | <name>    | <algo> <dir> <value> | Sets 6 character <i>IN</i> (bound) or <i>OUT</i> (bound) for <i>AUTH</i> (Manual Authentication) or <i>ESP</i> for <name> to <spi> (a hex value more than 0xFF) <value>.  |
|            | usepfs       | <name>    | <mode>               | Enables or disables Perfect Forward Secrecy for <name>.   |
|            | salife       | <name>    | <lifetime>           | Defines the name of the tunnel <name> the Security Association Life Time <300-65535> applies to in seconds.   |
|            | ike          | opmode    | <name> <opmode>      | Sets the Operation Mode of IKE for <name> to <i>Main</i> or <i>Aggressive</i> .   |
|            |              | myidtype  | <name> <idtype>      | Sets the Local ID type for IKE authentication for <name> (1 to 13 characters) to <idtype> ( <i>IP</i> , <i>FQDN</i> , or <i>UFQDN</i> ).  |
|            |              | remidtype | <name> <idtype>      | Sets the Remote ID type for IKE authentication for <name> (1 to 13 characters) to <idtype> ( <i>IP</i> , <i>FQDN</i> , or <i>UFQDN</i> ).   |



|           |        |            |  |
|-----------|--------|------------|--|
| myiddata  | <name> | <idtype>   | Sets the Local ID data for IKE authentication for <name> to <idtype>. This value is not required when the ID type is set to IP.                  |
| remiddata | <name> | <idtype>   | Sets the Local ID data for IKE authentication for <name> to <idtype>. This value is not required when the ID type is set to IP.                  |
| authtype  | <name> | <authtype> | Sets the IKE Authentication type for <name> to <authtype> ( <i>PSK</i> or <i>RSA</i> ).  |
| authalgo  | <name> | <authalgo> | Sets the IKE Authentication Algorithm for <name> to <i>MD5</i> or <i>SHA1</i> .  |
| phrase    | <name> | <phrase>   | Sets the IKE Authentication passphrase for <name> to <phrase>.   |
| encalgo   | <name> | <encalgo>  | Sets the IKE Encryption Algorithm for <name> to <encalgo> (one of <i>DES</i> , <i>3DES</i> , <i>AES128</i> , <i>AES192</i> , or <i>AES256</i> ). |
| lifetime  | <name> | <lifetime> | Sets the IKE Key life time in seconds for <name> to <lifetime>.  |
| group     | <name> | <group>    | Sets the IKE Diffie-Hellman Group for <name> to either <i>G768</i> or <i>G1024</i> .   |

**admin(network.wan.vpn)> delete**

Deletes VPN tunnel entries.

**Syntax**

**delete**    **all**            Deletes all VPN entries.  
              <name>        Deletes VPN entries by supplied name.

**Example**

```
admin(network.wan.vpn)>list
```

| Tunnel Name  | Type   | Remote IP/Mask   | Remote Gateway | Local WAN IP   |
|--------------|--------|------------------|----------------|----------------|
| Eng2EngAnnex | Manual | 192.168.32.2/24  | 192.168.33.1   | 192.168.24.198 |
| SJSharkey    | Manual | 206.107.22.45/27 | 206.107.22.2   | 209.235.12.55  |

```
admin(network.wan.vpn)>delete Eng2EngAnnex
```

```
admin(network.wan.vpn)>list
```

| Tunnel Name | Type   | Remote IP/Mask   | Remote Gateway | Local WAN IP  |
|-------------|--------|------------------|----------------|---------------|
| SJSharkey   | Manual | 206.107.22.45/27 | 206.107.22.2   | 209.235.12.55 |

```
admin(network.wan.vpn)>
```

## admin(network.wan.vpn)> list

Lists VPN tunnel entries.

### Syntax

**list** Lists all tunnel entries.  
**<name>** Lists detailed information about a specific tunnel <name>. Note that the <name> must match case with the name of the VPN tunnel entry.

### Example

```
admin(network.wan.vpn)>list
```

```
-----  
Tunnel Name   Type      Remote IP/Mask   Remote Gateway   Local WAN IP  
-----  
Eng2EngAnnex  Manual    192.168.32.2/24  192.168.33.1     192.168.24.198  
SJSharkey     Manual    206.107.22.45/27 206.107.22.2     209.235.12.55
```

```
admin(network.wan.vpn)>list SJSharkey
```

```
-----  
Detail listing of VPN entry:
```

```
-----  
Name           : SJSharkey  
Local Subnet    : 1  
Tunnel Type     : Manual  
Remote IP       : 206.107.22.45  
Remote IP Mask  : 255.255.255.224  
Remote Security Gateway : 206.107.22.2  
Local Security Gateway : 209.239.160.55  
AH Algorithm    : None  
Encryption Type : ESP  
Encryption Algorithm : DES  
ESP Inbound SPI : 0x00000100  
ESP Outbound SPI : 0x00000100
```

**admin(network.wan.vpn)> reset**

Resets all of the access point's VPN tunnels.

**Syntax**

**reset**                Resets all VPN tunnel states.

**Example**

```
admin(network.wan.vpn)>reset
```

```
VPN tunnels reset.
```

```
admin(network.wan.vpn)>
```

## **admin(network.wan.vpn)> stats**

Lists statistics for all active tunnels.

### **Syntax**

**stats**            Display statistics for all VPN tunnels.

### **Example**

```
admin(network.wan.vpn)>stats
```

| Tunnel Name  | Status     | SPI (OUT/IN) | Life Time | Bytes (Tx/Rx) |
|--------------|------------|--------------|-----------|---------------|
| Eng2EngAnnex | Not Active |              |           |               |
| SJSharkey    | Not Active |              |           |               |

**admin(network.wan.vpn)> ikestate**

Displays statistics for all active tunnels using an *Internet Key Exchange* (IKE).

**Syntax**

**ikestate**                Displays status about Internet Key Exchange (IKE) for all tunnels. In particular, the table indicates whether IKE is connected for any of the tunnels, it provides the destination IP address, and the remaining lifetime of the IKE key.

**Example**

```
admin(network.wan.vpn)>ikestate
```

| Tunnel Name  | IKE State     | Dest IP | Remaining Life |
|--------------|---------------|---------|----------------|
| Eng2EngAnnex | Not Connected | ----    | ---            |
| SJSharkey    | Not Connected | ----    | ---            |

```
admin(network.wan.vpn) >
```

## Network WAN Content Commands

### **admin(network.wan)>content**

Navigates to the Outbound Content Filtering menu. Content filtering allows system administrators to block specific commands and URL extensions from going out through the access point's WAN port. Therefore, content filtering affords system administrators selective control on the content proliferating the network and is a powerful data and network screening tool. Content filtering allows the blocking of up to 10 files or URL extensions and allows blocking of specific outbound HTTP, SMTP, and FTP requests. The items available under this command include:

|               |   |
|---------------|---|
| <b>addcmd</b> | Adds control commands to block outbound traffic.    |
| <b>delcmd</b> | Deletes control commands to block outbound traffic. |
| <b>list</b>   | Lists application control commands.                 |
| <b>..</b>     | Goes to the parent menu.                            |
| <b>/</b>      | Goes to the root menu.                              |
| <b>save</b>   | Saves the configuration to system flash.            |
| <b>quit</b>   | Quits the CLI.                                      |

**admin(network.wan.content)> addcmd**

Adds control commands to block outbound traffic.

**Syntax**

|               |      |         |   |
|---------------|------|---------|---|
| <b>addcmd</b> | web  |         | Adds WEB commands to block outbound traffic.  |
|               |      | proxy   | Adds a Web proxy command.                     |
|               |      | activex | Adds activex files.                           |
|               |      | file    | Adds Web URL extensions (10 files maximum)    |
|               | smtp |         | Adds SMTP commands to block outbound traffic. |
|               |      | helo    | helo command                                  |
|               |      | mail    | mail command                                  |
|               |      | rcpt    | rcpt command                                  |
|               |      | data    | data command                                  |
|               |      | quit    | quit command                                  |
|               |      | send    | send command                                  |
|               |      | saml    | saml command                                  |
|               |      | reset   | reset command                                 |
|               |      | vrfy    | vrfy command                                  |
|               |      | expn    | expn command                                  |
|               | ftp  |         | Adds FTP commands to block outbound traffic.  |
|               |      | put     | store command                                 |
|               |      | get     | retrieve command                              |
|               |      | ls      | directory list command                        |
|               |      | mkdir   | create directory command                      |
|               |      | cd      | change directory command                      |
|               |      | pasv    | passive mode command                          |

**Example**

```
admin(network.wan.content)>addcmd web proxy
admin(network.wan.content)>addcmd smtp data
admin(network.wan.content)>addcmd ftp put
```



## **admin(network.wan.content)> delcmd**

Deletes control commands to block outbound traffic.

### **Syntax**

|               |      |         |  |
|---------------|------|---------|--|
| <b>delcmd</b> | web  |         | Deletes WEB commands to block outbound traffic.  |
|               |      | proxy   | Deletes a Web proxy command.                     |
|               |      | activex | Deletes activex files.                           |
|               |      | file    | Deletes Web URL extensions (10 files maximum)    |
|               | smtp |         | Deletes SMTP commands to block outbound traffic. |
|               |      | helo    | helo command                                     |
|               |      | mail    | mail command                                     |
|               |      | rcpt    | rcpt command                                     |
|               |      | data    | data command                                     |
|               |      | quit    | quit command                                     |
|               |      | send    | send command                                     |
|               |      | saml    | saml command                                     |
|               |      | reset   | reset command                                    |
|               |      | vrfy    | vrfy command                                     |
|               |      | expn    | expn command                                     |
|               | ftp  |         | Deletes FTP commands to block outbound traffic.  |
|               |      | put     | store command                                    |
|               |      | get     | retrieve command                                 |
|               |      | ls      | directory list command                           |
|               |      | mkdir   | create directory command                         |
|               |      | cd      | change directory command                         |
|               |      | pasv    | passive mode command                             |

### **Example**

```
admin(network.wan.content)>delcmd web proxy
admin(network.wan.content)>delcmd smtp data
admin(network.wan.content)>delcmd ftp put
```

**admin(network.wan.content)> list**

Lists application control commands.

**Syntax**

|             |      |  |
|-------------|------|--|
| <b>list</b> | web  | Lists WEB application control record.  |
|             | smtp | Lists SMTP application control record. |
|             | ftp  | Lists FTP application control record.  |

**Example**

```
admin(network.wan.content)>list web
```

HTTP Files/Commands

```
Web Proxy           : deny
ActiveX             : allow
filename            :
```

```
admin(network.wan.content)>list smtp
```

SMTP Commands

```
HELO                : deny
MAIL                : allow
RCPT                : allow
DATA                : deny
QUIT                : allow
SEND                : allow
SAML                : allow
RESET               : allow
VRFY                : allow
EXPN                : allow
```

```
admin(network.wan.content)>list ftp
```

FTP Commands

```
Storing Files       : deny
Retreiving Files    : allow
Directory Files     : allow
Create Directory    : allow
Change Directory    : allow
Passive Operation   : allow
```

## Network WAN, Dynamic DNS Commands

### **admin(network.wan)> dyndns**

Navigates to the Dynamic DNS submenu. The items available under this command include:

|               |  |
|---------------|--|
| <b>set</b>    | Sets Dynamic DNS parameters.             |
| <b>update</b> | Sets key exchange parameters.            |
| <b>show</b>   | Shows the Dynamic DNS configuration.     |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |

**admin(network.wan.dyndns)> set**

Sets the access point's Dynamic DNS configuration.

**Syntax**

|            |          |                |  |
|------------|----------|----------------|--|
| <b>set</b> | mode     | enable/disable | Enables or disables the Dynamic DNS service for the access point.            |
|            | username | <name>         | Enter a 1 - 32 character username for the account used for the access point. |
|            | password | <password>     | Enter a 1 - 32 character password for the account used for the access point. |
|            | hostname | <host>         | Enter a 1 - 32 character hostname for the account used for the access point. |

**Example**

```
admin(network.wan.dyndns)>set mode enable
admin(network.wan.dyndns)>set username percival
admin(network.wan.dyndns)>set password mudskipper
admin(network.wan.dyndns)>set host greengiant
```

## **admin(network.wan.dyndns)> update**

Updates the access point's current WAN IP address with the DynDNS service.

### **Syntax**

**update**      Updates the access point's current WAN IP address with the DynDNS service (when DynDNS is enabled).

### **Example**

```
admin(network.wan.dyndns)>update
```

```
IP Address           : 157.235.91.231
Hostname             : greengiant
```

**admin(network.wan.dyndns)> show**

Shows the current Dynamic DNS configuration.

**Syntax**

**show**       Shows the access point's current Dynamic DNS configuration.

**Example**

```
admin(network.wan.dyndns)>show
```

**DynDNS Configuration**

|          |              |
|----------|--------------|
| Mode     | : enable     |
| Username | : percival   |
| Password | : *****      |
| Hostname | : greengiant |

**DynDNS Update Response**

|            |                  |
|------------|------------------|
| IP Address | : 157.235.91.231 |
| Hostname   | : greengiant     |
| Status     | : OK             |

## Network Wireless Commands

### **admin(network)> wireless**

Navigates to the access point wireless submenu. The items available under this command include:

|                       |   |
|-----------------------|---|
| <b>wlan</b>           | Displays the WLAN submenu used to create and configure up to 16 WLANs per access point.   |
| <b>security</b>       | Displays the security submenu used to create encryption and authentication based security policies for use with access point WLANs. |
| <b>acl</b>            | Displays to the <i>Access Control List</i> (ACL) submenu to restrict or allow client access to access point WLANs.                  |
| <b>radio</b>          | Displays the radio configuration submenu used to specify how the 802.11a or 802.11b/g radio is used with specific WLANs.            |
| <b>qos</b>            | Displays the <i>Quality of Service</i> (QoS) submenu to prioritize specific kinds of data traffic within a WLAN.                    |
| <b>bandwidth</b>      | Displays the Bandwidth Management submenu used to configure the order data is processed by an access point radio.                   |
| <b>rogue-ap</b>       | Displays the Rogue-AP submenu to configure devices located by the access point as friendly or threatening for interoperability.     |
| <b>wips</b>           | Goes to the WLAN Intrusion Prevention submenu.  |
| <b>mu-locationing</b> | Displays the Client locationing submenu.  |
| <b>..</b>             | Goes to the parent menu.  |
| <b>/</b>              | Goes to the root menu.  |
| <b>save</b>           | Saves the configuration to system flash.  |
| <b>quit</b>           | Quits the CLI.  |

## Network WLAN Commands

### **admin(network.wireless)> wlan**

Navigates to the access point wireless LAN (WLAN) submenu. The items available under this command include:

|               |   |
|---------------|---|
| <b>show</b>   | Displays the access point's current WLAN configuration. |
| <b>create</b> | Defines the parameters of a new WLAN.                   |
| <b>edit</b>   | Modifies the properties of an existing WLAN.            |
| <b>delete</b> | Deletes an existing WLAN.                               |
| <b>..</b>     | Goes to the parent menu.                                |
| <b>/</b>      | Goes to the root menu.                                  |
| <b>save</b>   | Saves the configuration to system flash.                |
| <b>quit</b>   | Quits the CLI.  |



## admin(network.wireless.wlan)> show

Displays the access point's current WLAN configuration.

### Syntax

|             |          |          |  |
|-------------|----------|----------|--|
| <b>show</b> | wlan     | <number> | Displays the configuration for the requested WLAN (WLAN 1 through 16). |
|             | security | <name>   | Displays the security policy for the WLAN (1-32).                      |
|             | acl      | <name>   | Displays the ACL policy used with the WLAN (1-32).                     |
|             | qos      | <name>   | Displays the name representing the QoS policy used with this WLAN.     |

### Example

```
admin(network.wireless.wlan)>show summary
```

```
WLAN1
WLAN Name           : Lobby
ESSID               : 101
Radio               : 11a, 11b/g
VLAN                :
Security Policy     : Default
QoS Policy          : Default
```

```
admin(network.wireless.wlan)>show wlan 1
```

```
ESS Identifier      : 101
WLAN Name           : Lobby
802.11a Radio       : available
802.11b/g Radio     : not available
Client Bridge Mesh Backhaul : available
Hotspot             : not available
Maximum MUs         : 127
MU Idle Timeout     : 30
Security Policy     : Default
MU Access Control   : Default
Kerberos User Name  : 101
Kerberos Password   : *****
Disallow MU to MU Communication : disable
Use Secure Beacon   : disable
Accept Broadcast ESSID : disable
QoS Policy          : Default
```

**admin(network.wireless.wlan)> create**

Navigates to the WLAN creation submenu.

**Syntax****create**

|      |            |                |   |
|------|------------|----------------|---|
| show | wlan       | <number>       | Displays newly created WLAN and policy number.  |
| set  | ess        | <ssid>         | Defines the ESSID for a target WLAN.  |
|      | wlan-name  | <name>         | Determines the name of this particular WLAN (1-32).   |
|      | 11a        | <mode>         | Enables or disables access to the access point 802.11a radio.   |
|      | 11bg       | <mode>         | Enables or disables access to the access point 802.11b/g radio.   |
|      | mesh       | <mode>         | Enables or disables the Client Bridge Mesh Backhaul option.   |
|      | hotspot    | <mode>         | Enables or disables the Hotspot mode.   |
|      | max-client | <number>       | Defines the maximum number of Clients able to operate within the WLAN (default = 127).  |
|      | security   | <name>         | Sets the security policy to the WLAN (1-32).  |
|      | acl        | <name>         | Sets the ACL policy to the WLAN (1-32).   |
|      | ipfilter   | <name>         | Sets the IP-Filtering Policy name.  |
|      | passwd     | <ascii string> | Defines a Kerberos password used if the WLAN's security policy uses a Kerberos server-based authentication scheme.              |
|      | no-mu-mu   | <mode>         | Enables or disables Clients associated to the same WLAN to not communicate with each other.                                     |
|      | sbeacon    | <mode>         | Enables or disables the access point from transmitting the ESSID in the beacon.   |
|      | bcast      | <mode>         | Enables or disables the access point from accepting broadcast IDs from Clients. Broadcast IDs are transmitted without security. |
|      | qos        | <name>         | Defines the index name representing the QoS policy used with this WLAN.   |
|      | add-wlan   |                | Apply the changes to the modified WLAN and exit.  |

**Example**

```
admin(network.wireless.wlan.create)>show wlan
```

```
ESS Identifier           :
WLAN Name               :
802.11a Radio           : available
802.11b/g Radio         : not available
Client Bridge Mesh Backhaul : not available
Hotspot                 : not available
Maximum MUs             : 127
MU Idle Timeout         : 30
Security Policy         : Default
MU Access Control       :
Kerberos User Name      : Default
Kerberos Password       : *****
Disallow MU to MU Communication : disable
Use Secure Beacon       : disable
Accept Broadcast ESSID  : disable
QoS Policy              : Default
```

```
admin(network.wireless.wlan.create)>show security
```

```
-----
```

| Secu Policy Name   | Authen | Encryption | Associated WLANs |
|--------------------|--------|------------|------------------|
| 1 Default          | Manual | no encrypt | Front Lobby      |
| 2 WEP Demo         | Manual | WEP 64     | 2nd Floor        |
| 3 Open             | Manual | no encrypt | 1st Floor        |
| WPA Countermeasure | enable |            |                  |

```
admin(network.wireless.wlan.create)>show acl
```

| ACL Policy Name | Associated WLANs |
|-----------------|------------------|
| 1 Default       | Front Lobby      |
| 2 Admin         | 3rd Floor        |
| 3 Demo Room     | 5th Floor        |

```
admin(network.wireless.wlan.create)>show qos
```

| QOS Policy Name | Associated WLANs |
|-----------------|------------------|
| 1 Default       | Front Lobby      |
| 2 Voice         | Audio Dept       |
| 3 Video         | Video Dept       |

The CLI treats the following as invalid characters, thus they should not be used in the creation of an ESSID (or other):

-> space < > | " & , \ ?

**admin(network.wireless.wlan)> edit**

Edits the properties of an existing WLAN policy.

**Syntax**

|             |       |  |
|-------------|-------|--|
| <b>edit</b> | <idx> | Edits the sequence number (index) in the WLAN summary. |
|-------------|-------|--|

## **admin(network.wireless.wlan)> delete**

Deletes an existing WLAN.

### **Syntax**

|               |             |   |
|---------------|-------------|---|
| <b>delete</b> | <wlan-name> | Deletes a target WLAN by name supplied. |
|               | all         | Deletes all WLAN configurations.        |

## Network Security Commands

### **admin(network.wireless)> security**

Navigates to the access point wireless security submenu. The items available under this command include:

|               |   |
|---------------|---|
| <b>show</b>   | Displays the access point's current security configuration. |
| <b>set</b>    | Sets security parameters.                                   |
| <b>create</b> | Defines the parameters of a security policy.                |
| <b>edit</b>   | Edits the properties of an existing security policy.        |
| <b>delete</b> | Removes a specific security policy.                         |
| <b>..</b>     | Goes to the parent menu.                                    |
| <b>/</b>      | Goes to the root menu.                                      |
| <b>save</b>   | Saves the configuration to system flash.                    |
| <b>quit</b>   | Quits the CLI.  |

## **admin(network.wireless.security)> show**

Displays the access point's current security configuration.

### **Syntax**

**show**      summary      Displays list of existing security policies (1-16).  
             policy      <id>      Displays the specified security policy <id>.

### **Example**

```
admin(network.wireless.security)>show summary
```

| -----            |        |            |                  |
|------------------|--------|------------|------------------|
| Secu Policy Name | Authen | Encryption | Associated WLANs |
| -----            |        |            |                  |
| 1 Default        | Manual | no encrypt | Lobby            |
| 2 WEP Demo       | Manual | WEP 64     | 2nd Floor        |
| 3 Open           | Manual | no encrypt | 1st Floor        |

WPA Countermeasure      enable

```
admin(network.wireless.security)>show policy 1
```

```
Policy Name                               : Default
Authentication                            : Manual Pre-shared key/No Authentication
Encryption type                           : no encryption
```

Related Commands:

**create**      Defines security parameters for the specified WLAN.

**admin(network.wireless.security)> create**

Defines the parameter of access point security policies.

**Syntax****create**

show

set        sec-name <name>

auth       <authtype>

kerb       realm       <name>  
server       <sidx>       <ip>

port       <sidx>       <port>

eap       server       <sidx>       <ip>

port       <sidx>       <port>

secret       <sidx>       <secret>

reauth       mode       <mode>

period       <time>

retry       <number>

accounting   mode       <mode>

server       <ip>

port       <port>

secret       <secret>

timeout       <period>

Defines the parameters of a security policy.

Displays new or existing security policy parameters.

Sets the name of the security policy.

Sets the authentication type for WLAN <idx> to <type> (*none*, *eap*, or *kerberos*).

Note: Kerberos parameters are only in affect if "kerberos" is specified for the authentication method (set auth <type>).

Sets the Kerberos realm.

Sets the Kerberos server <sidx> (1-primary, 2-backup, or 3-remote) to KDC IP address.

Sets the Kerberos port to <port> (KDC port) for server <ksidx> (1-primary, 2-backup, or 3-remote).

Note: EAP parameters are only in affect if "eap" is specified for the authentication method (set auth <type>).

Sets the radius server (1-primary or as 2-secondary) IP address <ip>.

Sets the radius server <sidx> (1-primary or 2-secondary) <port> (1-65535).

Sets the EAP shared secret <secret> (1-63 characters) for server <sidx> (1-primary or 2-secondary).

*The default password is "admin123".*

Enables or disables EAP reauthentication.

Sets the reauthentication period <period> in seconds (30-9999).

Sets the maximum number of reauthentication retries <retry> (1-99).

Enable or disable Radius accounting.

Set external Radius server IP address.

Set external Radius server port number.

Set external Radius server shared secret password.

Defines the Client timeout period in seconds (1-255).



|  |              |                 |               |   |
|--|--------------|-----------------|---------------|---|
|  |              | retry           | <number>      | Sets the maximum number of retries to <retry> (1-10).   |
|  |              | syslog          | <mode>        | Enable or disable syslog messages.  |
|  |              | ip              | <ip>          | Defines syslog server IP address.   |
|  | adv          | mu-quiet        | <time>        | Set the EAP MU/supplicant quiet period to <time> seconds (1-65535).   |
|  |              | mu-timeout      | <timeout>     | Sets the EAP MU/supplicant timeout in seconds (1-255).  |
|  |              | mu-tx           | <time>        | Sets the EAP MU/supplicant TX period <time> in seconds (1-65535).   |
|  |              | mu-retry        | <count>       | Sets the EAP maximum number of MU retries to <count> (1-10).  |
|  |              | svr-timeout     | <time>        | Sets the server timeout <time> in seconds (1-255).  |
|  |              | svr-retry       | <count>       | Sets the maximum number of server retries to <count> (1-255).<br>Note: The WEP authentication mechanism saves up to four different keys (one for each WLAN). It is not requirement to set all keys, but you must associate a WLAN with the same keys. |
|  | enc          | <idx>           | <type>        | Sets the encryption type to <type> (one of <i>none</i> , <i>wep40</i> , <i>wep104</i> , <i>keyguard</i> , <i>tkip</i> , or <i>ccmp</i> ) for WLAN <idx>.  |
|  | wep-keyguard | passkey         | <passkey>     | The passkey used as a text abbreviation for the entire key length (4-32).   |
|  |              | index           | <key index>   | Selects the WEP/KeyGuard key (from one of the four potential values of <key index> (1-4).   |
|  |              | hex-key         | <kidx>        | <key string> Sets the WEP/KeyGuard key for key index <kidx> (1-4) for WLAN <kidx> to <key string>.  |
|  |              | ascii-key       | <kidx>        | <key string> Sets the WEP/KeyGuard key for key index <kidx> (1-4) for WLAN <kidx> to <key string>.  |
|  |              | mixed-mode      | <mode>        | Enables or disables interoperation with WEP128 clients.<br>Note: TKIP parameters are only affected if "tkip" is selected as the encryption type.  |
|  | tkip         | rotate-mode     | <mode>        | Enables or disabled the broadcast key.  |
|  |              | interval        | <time>        | Sets the broadcast key rotation interval to <time> in seconds (300-604800).   |
|  |              | allow-wpa2-tkip | <mode>        | Enables or disables the interoperation with wpa2-tkip clients.  |
|  |              | preauth         | <mode>        | Enables or disables preauthentication (fast roaming).   |
|  |              | type            | <key type>    | Sets the TKIP key type.   |
|  |              | key             | <256 bit key> | Sets the TKIP key to <256 bit key>.   |

|            |             |                |   |
|------------|-------------|----------------|---|
|            | phrase      | <ascii phrase> | Sets the TKIP ASCII pass phrase to <ascii phrase> (8-63 characters).        |
| ccmp       | rotate-mode | <mode>         | Enables or disabled the broadcast key.                                      |
|            | interval    | <time>         | Sets the broadcast key rotation interval to <time> in seconds (300-604800). |
|            | type        | <key type>     | Sets the CCMP key type.   |
|            | phrase      | <ascii phrase> | Sets the CCMP ASCII pass phrase to <ascii phrase> (8-63 characters).        |
|            | key         | <256 bit key>  | Sets the CCMP key to <256 bit key>.   |
|            | mixed-mode  | <mode>         | Enables or disables mixed mode (allowing WPA-TKIP clients).                 |
|            | preauth     | <mode>         | Enables or disables preauthentication (fast roaming).                       |
| add-policy |             |                | Adds the policy and exits.  |
| ..         |             |                | Disregards the policy creation and exits the CLI session.                   |

## **admin(network.wireless.security)> edit**

Edits the properties of a specific security policy.

### **Syntax**

**edit**      <idx>      Edits a profile specified by its ID.

A new context opens for the profile being edited.

AP35xx>admin(network.wireless.security.edit)>

## Network Security Policy Edit Commands

### **admin(network.wireless.security)> edit**

Navigates to the access point wireless security policy edit submenu. The items available under this menu include:

|               |   |
|---------------|---|
| <b>show</b>   | Displays the security policy parameters for the selected security policy. |
| <b>set</b>    | Sets security parameters for the selected policy.                         |
| <b>change</b> | Changes the policy and exits this submenu.                                |
| <b>..</b>     | Goes to the parent menu.  |

**admin(network.wireless.security.edit)> show**

### **Description:**

Displays the security policy details for the selected policy.

### **Syntax :**

**show**                      Displays the new or modified security policy parameters.

### **Example**

```
admin(network.wireless.security.edit)>show
```

```
Policy Name                : Default
Authentication type        : Manual Pre-shared key / No authentication

Encryption type            : WPA/TKIP
  ccmp broadcast key rotate mode : disable
  ccmp key type              : phrase
  ccmp phrase                : *****
  ccmp mixed mode (allow WPA)  : disable
  tkip broadcast key rotate mode : disable
  tkip key type              : key
  tkip key                   : *****
  allow wpa2 tkip             : enable
```

**admin(network.wireless.security.edit)> set****Description:**

Configures the different parameters for the selected security policy.

**Syntax**

|            |          |             |                 |   |
|------------|----------|-------------|-----------------|---|
| <b>set</b> | sec-name | <name-str>  |                 | Sets the name of the selected security profile to <name-str>.   |
|            | auth     | <auth-type> |                 | Sets the authentication type for the selected security profile to <auth-type> ( <i>none, eap, kerberos</i> ).   |
|            | kerb     |             |                 |   |
|            |          | realm       | <name-str>      | Sets the Kerberos realm name to <name-str>.   |
|            |          | server      | <s-idx> <ip>    | Sets the Kerberos server type to <s-idx> ( <i>1 - primary, 2 - backup, 3 - remote</i> ). Also sets the IP address of the server to <ip>.              |
|            |          | port        | <s-idx> <p-num> | Sets the Kerberos server port to <p-num> ( <i>1-65535</i> ) for the server type <s-idx> ( <i>1 - primary, 2 - backup, 3 - remote</i> ).               |
|            | eap      |             |                 |   |
|            |          | server      | <s-idx> <ip>    | Sets the RADIUS Server type to <s-idx> ( <i>1 - primary, 2 - secondary</i> ) and sets its IP address to <ip>.   |
|            |          | port        | <s-idx> <p-num> | Sets the RADIUS Server port number for server type <s-idx> ( <i>1 - primary, 2 - secondary</i> ) to port number <p-num> ( <i>1-65535</i> ).           |
|            |          | secret      | <s-idx> <c>     | Sets the shared secret for the RADIUS Server type <s-idx> ( <i>1 - primary, 2 - secondary</i> ) to a character string <c> ( <i>1-127</i> characters). |
|            |          | reauth      |                 |   |
|            |          |             | mode            | <mode> Enables or disables EAP reauthentication.  |
|            |          |             | period          | <time> Sets the EAP reauthentication period to <time> ( <i>30-9999</i> seconds)   |
|            |          |             | retry           | <num> Sets the EAP reauthentication retry count to <num> ( <i>1-99</i> )  |
|            |          | accounting  |                 |   |
|            |          |             | mode            | <mode> Enables or disables RADIUS Accounting.   |
|            |          |             | server          | <ip-addr> Sets the IP of the external RADIUS Accounting Server.   |
|            |          |             | port            | <p> Sets the port for the external RADIUS Accounting Server.  |
|            |          |             | secret          | <c> Sets the common shared secret for RADIUS Accounting.  |
|            |          |             | timeout         | <time> Sets the timeout period to <time> ( <i>1-255</i> seconds).   |
|            |          |             | retry           | <num> Sets the retry count to <num> ( <i>1-10</i> ).  |
|            |          |             | syslog          | <mode> Enables or disables syslog mode.   |
|            |          |             | ip              | <ip-addr> Sets the IP address of the syslog server.   |
|            |          | adv         |                 |   |
|            |          |             | mu-quiet        | <time> Sets supplicant Quiet period to <time> ( <i>1-65535</i> seconds).  |
|            |          |             | mu-timeout      | <timeout> Sets supplicant Timeout period to <timeout> ( <i>1-255</i> seconds).  |
|            |          |             | mu-tx           | <time> Sets supplicant Tx period to <time> ( <i>1-65535</i> seconds).   |
|            |          |             | mu-retry        | <count> Sets max retries to <count> ( <i>1-10</i> ).  |

|              |            |                 |                   |   |
|--------------|------------|-----------------|-------------------|---|
| enc          | <enc-type> | svr-timeout     | <timeout>         | Sets server timeout to <timeout> (1-255 seconds).   |
|              |            | svr-retry       | <count>           | Sets server max retries to <count> (1-255).   |
| wep-keyguard |            |                 |                   | Sets the encryption type to <enc-type> (none, wep40, wep104, keyguard, tkip, ccmp)  |
|              |            |                 |                   | For Manual pre-shared key or no authentication only.  |
|              |            | passkey         | <pass-key>        | Sets the WEP/Keyguard-MCM passkey to <passkey> (4-32 chars).  |
|              |            | index           | <key-idx>         | Sets the WEP/Keyguard-MCM key index to <key-idx> (1-4).   |
|              |            | hex-key         | <k-idx> <key-str> | Sets the Hexadecimal key <key-str> for the key index <key-idx> (1-4). <key-str> can be 10 hex digits for WEP40 and 26 digits for WEP104/Keyguard. |
| tkip         |            | ascii-key       | <k-idx> <key-str> | Sets the ASCII key <key-str> for the key index <key-idx> (1-4). <key-str> can be 5 chars for WEP40 and 13 chars for WEP104/Keyguard.              |
|              |            | mixed-mode      | <mode>            | Enables or disables Allow WEP128 clients.   |
|              |            | rotate-mode     | <mode>            | Enables or disables Broadcast Key Rotation.   |
|              |            | interval        | <time>            | Sets Broadcast Key Rotation interval to <time> (30-604800 seconds).   |
|              |            | allow-wpa2-tkip | <mode>            | Enables or disables WPA2/TKIP.  |
|              |            | preauth         | <mode>            | Enables or disables preauthentication.  |
| ccmp         |            | type            | <key-type>        | Sets TKIP key type to <key-type> (phrase, key)  |
|              |            | key             | <256-bit-key>     | Sets the 256-bit TKIP key to <256-bit-key> (64 hex digits).   |
|              |            | phrase          | <ascii-phrase>    | Sets the ASCII TKIP key to <ascii-phrase> (8-63 characters).  |
|              |            | rotate-mode     | <mode>            | Enables or disables Broadcast Key Rotation.   |
|              |            | interval        | <time>            | Sets Broadcast Key Rotation interval to <time> (30-604800 seconds).   |
|              |            | type            | <key-type>        | Sets CCMP key type to <key-type> (phrase, key)  |
|              |            | phrase          | <ascii-phrase>    | Sets the ASCII CCMP key to <ascii-phrase> (8-63 chars).   |
|              |            | key             | <256-bit-key>     | Sets the 256-bit CCMP key to <256-bit-key> (64 hex digits).   |
|              |            | mixed-mode      | <mode>            | Enables or disables mixed-mode operation.   |
|              |            | preauth         | <mode>            | Enables or disables preauthentication.  |

## Example

```

admin(network.wireless.security)>edit 1
admin(network.wireless.security.edit)>show
Policy Name                : Default
Authentication type        : Manual Pre-shared key / No authentication

Encryption type            : WPA/TKIP
ccmp broadcast key rotate mode : disable
ccmp key type              : phrase
ccmp phrase                : *****
ccmp mixed mode (allow WPA) : disable
tkip broadcast key rotate mode : disable
tkip key type              : key

```

```

tkip key                               : *****
allow wpa2 tkip                        : enable

admin(network.wireless.security.edit)>set auth none
admin(network.wireless.security.edit)>set enc tkip
admin(network.wireless.security.edit)>set tkip rotate-mode enable
admin(network.wireless.security.edit)>set tkip interval 46
admin(network.wireless.security.edit)>show

Policy Name                           : Default
Authentication type                    : Manual Pre-shared key / No authentication

Encryption type                       : WPA/TKIP
  ccmp broadcast key rotate mode      : disable
  ccmp key type                       : key
  ccmp key                            :
101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F
  ccmp mixed mode (allow WPA)         : disable
  tkip broadcast key rotate mode       : enable
  update broadcast keys every         : 46 (30-604800) seconds
  tkip key type                       : key
  tkip key                            :
101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F
  allow wpa2 tkip                     : enable

```



## **admin(network.wireless.security.edit)> change**

### **Description:**

Saves the policy changes and exits to the security submenu.

### **Syntax**

**change** Saves the policy changes and exists to the security submenu.

### **Example**

```
admin(network.wireless.security.edit)>set auth none
admin(network.wireless.security.edit)>set enc tkip
admin(network.wireless.security.edit)>set tkip rotate-mode enable
admin(network.wireless.security.edit)>set tkip interval 46
admin(network.wireless.security.edit)>show
```

```
Policy Name                : Default
Authentication type         : Manual Pre-shared key / No authentication
```

```
Encryption type            : WPA/TKIP
  ccmp broadcast key rotate mode : disable
  ccmp key type               : key
  ccmp key                    :
101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F
  ccmp mixed mode (allow WPA)  : disable
  tkip broadcast key rotate mode : enable
  update broadcast keys every  : 46 (30-604800) seconds
  tkip key type               : key
  tkip key                    :
101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D2E2F
  allow wpa2 tkip              : enable
admin(network.wireless.security.edit)>change
```

**admin(network.wireless.security)> delete**

Deletes a specific security policy.

**Syntax**

|               |            |  |
|---------------|------------|--|
| <b>delete</b> | <sec-name> | Removes the specified security policy from the list of supported policies. |
|               | <all>      | Removes all security policies except the default policy.                   |

## Network ACL Commands

### **admin(network.wireless.acl)>**

Navigates to the access point *Access Control List* (ACL) submenu. The items available under this command include:

|               |  |
|---------------|--|
| <b>show</b>   | Displays the access point's current ACL configuration. |
| <b>create</b> | Creates a Client ACL policy.                           |
| <b>edit</b>   | Edits the properties of an existing Client ACL policy. |
| <b>delete</b> | Removes an Client ACL policy.                          |
| <b>..</b>     | Goes to the parent menu.                               |
| <b>/</b>      | Goes to the root menu.                                 |
| <b>save</b>   | Saves the configuration to system flash.               |
| <b>quit</b>   | Quits the CLI.   |

**admin(network.wireless.acl)> show**

Displays the access point's current ACL configuration.

**Syntax**

|             |                |  |
|-------------|----------------|--|
| <b>show</b> | summary        | Displays the list of existing Client ACL policies. |
|             | policy <index> | Displays the requested Client ACL index policy.    |

**Example**

```
admin(network.wireless.acl)>show summary
```

| ACL Policy Name | Associated WLANs   |
|-----------------|--------------------|
| 1 Default       | Front Lobby, WLAN1 |
| 2 Admin         | Administration     |
| 3 Demo Room     | Customers          |

```
admin(network.wireless.acl)>show policy 1
```

```
Policy Name           : Default
Policy Mode           : allow
```

| index | start mac    | end mac      |
|-------|--------------|--------------|
| 1     | 00A0F8348787 | 00A0F8348798 |

## admin(network.wireless.acl)> create

Creates a Client ACL policy.

### Syntax

|               |            |                            |            |   |
|---------------|------------|----------------------------|------------|---|
| <b>create</b> | show       |                            | <acl-name> | Displays the parameters of a new ACL policy.  |
|               | set        | acl-name                   | <index>    | Sets the Client ACL policy name.  |
|               |            | mode                       | <acl-mode> | Sets the ACL mode for the defined index (1-16). Allowed Clients can access the access point managed LAN. Options are <i>deny</i> and <i>allow</i> . |
|               | add-addr   | <mac1> or<br><mac1> <mac2> |            | Adds specified MAC address to list of ACL MAC addresses.  |
|               | delete     | <index>                    | <all>      | Removes either a specified ACL index or all ACL entries.  |
|               | add-policy |                            |            | Completes the policy creation and exits the CLI.  |
|               | ..         |                            |            | Cancels the creation of the ACL and exits the CLI.  |

### Example

```
admin(network.wireless.acl.create)>show
```

```
Policy Name           : Front Lobby
Policy Mode           : allow
```

```
-----
index      start mac      end mac
-----
1          00A0F8334455    00A0F8334455
2          00A0F8400000    00A0F8402001
```

```
admin(network.wireless.acl.create)>set acl-name engineering
admin(network.wireless.acl.create)>set mode deny
admin(network.wireless.acl.create)>add-addr 00A0F843AABB
admin(network.wireless.acl.create)>add-policy
```

**admin(network.wireless.acl.edit)>**

Edits the properties of an existing Client ACL policy.

**Syntax**

|                 |  |
|-----------------|--|
| <b>show</b>     | Displays Client ACL policy and its parameters.                                       |
| <b>set</b>      | Modifies the properties of an existing Client ACL policy.                            |
| <b>add-addr</b> | Adds an Client ACL table entry.  |
| <b>delete</b>   | Deletes an Client ACL table entry, including starting and ending MAC address ranges. |
| <b>change</b>   | Completes the changes made and exits the session.                                    |
| <b>..</b>       | Cancels the changes made and exits the session.                                      |

## **admin(network.wireless.acl)> delete**

Removes an Client ACL policy.

### **Syntax**

|               |            |  |
|---------------|------------|--|
| <b>delete</b> | <acl name> | Deletes a specific Client ACL policy.                            |
|               | all        | Deletes all Client ACL policies (except for the default policy). |

## Network Radio Configuration Commands

### **admin(network.wireless)> radio**

Navigates to the access point Radio submenu. The items available under this command include:

|               |   |
|---------------|---|
| <b>show</b>   | Summarizes access point radio parameters at a high-level. |
| <b>set</b>    | Defines the access point radio configuration.             |
| <b>radio1</b> | Displays the 802.11b/g radio submenu.                     |
| <b>radio2</b> | Displays the 802.11a radio submenu.                       |
| <b>..</b>     | Goes to the parent menu.                                  |
| <b>/</b>      | Goes to the root menu.                                    |
| <b>save</b>   | Saves the configuration to system flash.                  |
| <b>quit</b>   | Quits the CLI.  |



## **admin(network.wireless.radio)> show**

Displays the access point's current radio configuration.

### **Syntax**

**show**      Displays the access point's current radio configuration.

### **Example**

```
admin(network.wireless.radio)>show
```

#### Radio Configuration

##### Radio 1

|                      |                       |
|----------------------|-----------------------|
| Name                 | : Radio 1             |
| Radio Mode           | : enable              |
| RF Band of Operation | : 802.11b/g (2.4 GHz) |
| RF Function          | : WLAN                |

##### Wireless Mesh Configuration:

|                         |           |
|-------------------------|-----------|
| Base Bridge Mode        | : enable  |
| Max Wireless AP Clients | : 6       |
| Client Bridge Mode      | : disable |
| Client Bridge WLAN      | : WLAN1   |
| Mesh Connection Timeout | : enable  |

##### Radio 2

|                      |                   |
|----------------------|-------------------|
| Name                 | : Radio 2         |
| Radio Mode           | : enable          |
| RF Band of Operation | : 802.11a (5 GHz) |
| RF Function          | : WLAN            |

##### Wireless Mesh Configuration:

|                         |           |
|-------------------------|-----------|
| Base Bridge Mode        | : enable  |
| Max Wireless AP Clients | : 5       |
| Client Bridge Mode      | : disable |
| Client Bridge WLAN      | : WLAN1   |
| Mesh Connection Timeout | : enable  |

|                      |                    |
|----------------------|--------------------|
| Dot11 Auth Algorithm | : open-system-only |
|----------------------|--------------------|

**admin(network.wireless.radio)> set**

Enables an access point Radio and defines the RF band of operation.

**Syntax**

|            |              |                  |   |
|------------|--------------|------------------|---|
| <b>set</b> | 11a          | <mode>           | Enables or disables the access point's 802.11a radio.   |
|            | 11bg         | <mode>           | Enables or disables the access point's 802.11b/g radio.   |
|            | rf-function  | <mode>           | Sets the WLAN or WIPS sensor mode for the specified radio index <idx>.  |
|            | mesh-base    | <mode>           | Enables or disables base bridge mode.   |
|            | mesh-max     |                  | Sets the maximum number of wireless bridge clients.   |
|            | mesh-client  | <mode>           | Enables or Disables client bridge mode.   |
|            | mesh-timeout | <period>         | Sets the client bridge link timeout for the radio index..   |
|            | mesh-wlan    | <name>           | Defines the client bridge WLAN name.  |
|            | dot11-auth   | <auth-algorithm> | Defines dot11 level authentication algorithm to either <i>open-system-only</i> or <i>shared-key-allowed</i> . |

**Example**

```
admin(network.wireless.radio)>set 11a disable
admin(network.wireless.radio)>set 11bg enable
admin(network.wireless.radio)>set rf-function 1 wlan
admin(network.wireless.radio)>set mesh-base enable
admin(network.wireless.radio)>set mesh-max 11
admin(network.wireless.radio)>set mesh-client disable
admin(network.wireless.radio)>set mesh-timeout 1 45
admin(network.wireless.radio)>set mesh-wlan wlan1
admin(network.wireless.radio)>set dot11-auth shared-key-allowed
admin(network.wireless.radio)>show
```

## Radio Configuration

```
Radio 1
Name : Radio 1
Radio Mode : enable
RF Band of Operation : 802.11b/g (2.4 GHz)
```

## Wireless AP Configuration:

```
Base Bridge Mode : enable
Max Wireless AP Clients : 11
Client Bridge Mode : disable
Client Bridge WLAN : WLAN1
Mesh Connection Timeout : 45 sec.
```

```
Dot11 Auth Algorithm : shared-key-allowed
```

## **admin(network.wireless.radio)> radio1**

Navigates to a 802.11b/g radio specific submenu. The items available under this command include:

### **Syntax**

|                 |   |
|-----------------|---|
| <b>show</b>     | Displays 802.11b/g radio settings.                          |
| <b>set</b>      | Defines specific 802.11b/g radio parameters.                |
| <b>delete</b>   | Deletes the channels defined within the ACS exception list. |
| <b>advanced</b> | Displays the Advanced radio settings submenu.               |
| <b>mesh</b>     | Goes to the Wireless AP Connections submenu.                |
| <b>..</b>       | Goes to the parent menu.                                    |
| <b>/</b>        | Goes to the root menu.                                      |
| <b>save</b>     | Saves the configuration to system flash.                    |
| <b>quit</b>     | Quits the CLI.  |

**admin(network.wireless.radio.radio1)> show**

Displays specific 802.11b/g radio settings.

**Syntax**

**show**                    radio                    Displays specific 802.11b/g radio settings.  
                              qos                        Displays specific 802.11b/g radio WMM QoS settings.

**Example**

```
admin(network.wireless.radio.radio1)>show radio
```

## Radio Setting Information

```
Placement                : indoor
MAC Address               : 00A0F8715920
Radio Type                : 802.11b/g
ERP Protection            : Off

Channel Setting           : user selection
ACS Exception Channel List :
Antenna Diversity         : full
Power Level               : 5 dbm (4 mW)

802.11b/g mode            : B-Only
Basic Rates               : 1 2 5.5 11
Supported Rates           : 1 2 5.5 11

Beacon Interval           : 100 K-usec
DTIM Interval per BSSID
    1                     : 10 beacon intvls
    2                     : 10 beacon intvls
    3                     : 10 beacon intvls
    4                     : 10 beacon intvls

short preamble            : disable
RTS Threshold             : 2341 bytes
Extended Range            : 0 miles
```

```
QBSS Channel Util Beacon Intvl : 10 beacon intvls
QBSS Load Element Mode         : enable
admin(network.wireless.radio.radio1)>show qos
```

## Radio QOS Parameter Set                    11g-default

| Access Category | CWMin | CWMax | AIFSN | TXOPs (32 usec) | TXOPs ms |
|-----------------|-------|-------|-------|-----------------|----------|
| Background      | 15    | 1023  | 7     | 0               | 0.000    |
| Best Effort     | 15    | 63    | 3     | 31              | 0.992    |
| Video           | 7     | 15    | 1     | 94              | 3.008    |
| Voice           | 3     | 7     | 1     | 47              | 1.504    |

**CAUTION**

*If you do NOT include the index number (for example, "set dtim 50"), the DTIMs for all four BSSIDs will be changed to 50. To change individual DTIMs for BSSIDs, specify the BSS Index number (for example, "set dtim 2 50"). This will change the DTIM for BSSID 2 to 50.*

## admin(network.wireless.radio.802-11bg)> set

Defines specific 802.11b/g radio parameters.

### Syntax

|            |                    |   |
|------------|--------------------|---|
| <b>set</b> | placement          | Defines the access point radio placement as indoors or outdoors.                        |
|            | ch-mode            | Determines how the radio channel is selected.   |
|            | channel            | Defines the actual channel used by the radio.   |
|            | acs-exception-list | Sets the ACS exception list (for auto selection only) for up to 3 channels.             |
|            | antenna            | Sets the radio antenna power  |
|            | power              | Defines the radio antenna power transmit level.   |
|            | bg-mode            | Enables or disables 802-11bg radio mode support.  |
|            | rates              | Sets the supported radio transmit rates.  |
|            | beacon             | Sets the beacon interval used by the radio.   |
|            | dtim               | Defines the DTIM interval (by index) used by the radio.                                 |
|            | preamble           | Enables or disables support for short preamble for the radio.                           |
|            | rts                | Defines the RTS Threshold value for the radio.  |
|            | range              | Sets the radio's extended range (in miles 0-50).  |
|            | qos                | Defines the cwmin, cwmax, aifsn and txops levels for the QoS policy used for the radio. |
|            | qbss-beacon        | Sets the QBSS Channel Util Beacon Interval in kilo-usec (10 - 200).                     |
|            | qbss-mode          | Enables/disables the QBSS load element.   |

### Example

```
admin(network.wireless.radio.802-11bg)>set placement indoor
admin(network.wireless.radio.802-11bg)>set ch-mode user
admin(network.wireless.radio.802-11bg)>set channel 1
admin(network.wireless.radio.802-11bg)>set acs-exception-list 10
admin(network.wireless.radio.802-11bg)>set antenna full
admin(network.wireless.radio.802-11bg)>set power 4
admin(network.wireless.radio.802-11bg)>set bg-mode enable
admin(network.wireless.radio.802-11bg)>set rates
admin(network.wireless.radio.802-11bg)>set beacon 100
admin(network.wireless.radio.802-11bg)>set dtim 1 40
admin(network.wireless.radio.802-11bg)>set preamble disable
admin(network.wireless.radio.802-11bg)>set rts 2341
admin(network.wireless.radio.802-11bg)>set qos cwmin 125
admin(network.wireless.radio.802-11bg)>set qos cwmax 255
admin(network.wireless.radio.802-11bg)>set qos aifsn 7
admin(network.wireless.radio.802-11bg)>set qos txops 0
admin(network.wireless.radio.802-11bg)>set qbss-beacon 110
admin(network.wireless.radio.802-11bg)>set qbss-mode enable
```



### CAUTION

*If you do NOT include the index number (for example, "set dtim 50"), the DTIMs for all four BSSIDs will be changed to 50. To change individual DTIMs for BSSIDs, specify the BSS Index number (for example, "set dtim 2 50"). This will change the DTIM for BSSID 2 to 50.*

**admin(network.wireless.radio.802-11bg)> advanced**

Displays the advanced submenu for the 802.11b/g radio. The items available under this command include:

**Syntax**

|             |   |
|-------------|---|
| <b>show</b> | Displays advanced radio settings for the 802.11b/g radio. |
| <b>set</b>  | Defines advanced parameters for the 802.11b/g radio.      |
| <b>..</b>   | Goes to the parent menu.                                  |
| <b>/</b>    | Goes to the root menu.                                    |
| <b>save</b> | Saves the configuration to system flash.                  |
| <b>quit</b> | Quits the CLI.  |

## admin(network.wireless.radio.802-11bg.advanced)> show

Displays the BSSID to WLAN mapping for the 802.11b/g radio.

### Syntax

**show**                      advanced      Displays advanced settings for the 802.11b/g radio.  
                             wlan            Displays WLAN summary list for the 802.11b/g radio.

### Example

```
admin(network.wireless.radio.802-11bg.advanced)>show advanced
```

| WLAN   | BSS ID | BC/MC Cipher | Status | Message             |
|--------|--------|--------------|--------|---------------------|
| Lobby  | 1      | Open         | good   | configuration is ok |
| HR     | 2      | Open         | good   | configuration is ok |
| Office | 3      | Open         | good   | configuration is ok |

  

| BSSID | Primary WLAN |
|-------|--------------|
| 1     | Lobby        |
| 2     | HR           |
| 3     | Office       |

```
admin(network.wireless.radio.802-11bg.advanced)>show wlan
```

```
WLAN 1:
WLAN name      : WLAN1
ESS ID         : 101
Radio          : 11a,11b/g
VLAN           : <none>
Security Policy: Default
QoS Policy     : Default
```

**admin(network.wireless.radio.802-11bg.advanced)> set**

Defines advanced parameters for the target 802.11b/g radio.

**Syntax**

|            |      |             |             |  |
|------------|------|-------------|-------------|--|
| <b>set</b> | wlan | <wlan-name> | <bssid>     | Defines advanced WLAN to BSSID mapping for the target radio. |
|            | bss  | <bss-id>    | <wlan name> | Sets the BSSID to primary WLAN definition.                   |

**Example**

```
admin(network.wireless.radio.802-11bg.advanced)>set wlan demoroom 1
admin(network.wireless.radio.802-11bg.advanced)>set bss 1 demoroom
```



## **admin(network.wireless.radio)> radio2**

Navigates to a 802.11a specific radio submenu. The items available under this command include:

### **Syntax**

|                 |   |
|-----------------|---|
| <b>show</b>     | Displays 802.11a radio settings               |
| <b>set</b>      | Defines specific 802.11a radio parameters.    |
| <b>delete</b>   | Deletes the ACS exception channels.           |
| <b>advanced</b> | Displays the Advanced radio settings submenu. |
| <b>mesh</b>     | Goes to the Wireless AP Connections submenu.  |
| <b>..</b>       | Goes to the parent menu.                      |
| <b>/</b>        | Goes to the root menu.                        |
| <b>save</b>     | Saves the configuration to system flash.      |
| <b>quit</b>     | Quits the CLI.                                |

**admin(network.wireless.radio.802-11a)> show**

Displays specific 802.11a radio settings.

**Syntax**

**show**                      radio                      Displays 802.11a radio settings.  
                                  qos                      Displays 802.11a radio WMM QoS settings.

**Example**

```
admin(network.wireless.radio.802-11a)>show radio
```

## Radio Setting Information

```
Placement                : indoor
MAC Address               : 00A0F8715920
Radio Type                : 802.11a

Channel Setting           : user selection
ACS Exception Channel List : 44 153 161
Antenna Diversity         : full
Power Level               : 5 dbm (4 mW)

Basic Rates               : 6 12 24
Supported Rates           : 6 9 12 18 24 36 48 54

Beacon Interval           : 100 K-usec
DTIM Interval per BSSID
    1                     : 10 beacon intvls
    2                     : 10 beacon intvls
    3                     : 10 beacon intvls
    4                     : 10 beacon intvls

RTS Threshold             : 2341 bytes
Extended Range            : 0 miles

QBSS Channel Util Beacon Intvl : 10 beacon intvls
QBSS Load Element Mode     : enable
```

```
admin(network.wireless.radio.802-11a)>show qos
```

Radio QOS Parameter Set:                      11a default

| Access Category | CWMin | CWMax | AIFSN | TXOPs (32 sec) | TXOPs ms |
|-----------------|-------|-------|-------|----------------|----------|
| Background      | 15    | 1023  | 7     | 0              | 0.000    |
| Best Effort     | 15    | 63    | 3     | 31             | 0.992    |
| Video           | 7     | 15    | 1     | 94             | 3.008    |
| Voice           | 3     | 7     | 1     | 47             | 1.504    |

## **admin(network.wireless.radio.802-11a)> set**

Defines specific 802.11a radio parameters.

### **Syntax**

|            |                    |   |
|------------|--------------------|---|
| <b>set</b> | placement          | Defines the access point radio placement as indoors or outdoors.                        |
|            | ch-mode            | Determines how the radio channel is selected.   |
|            | channel            | Defines the actual channel used by the radio.   |
|            | acs-exception-list | Used to define the automatic channel selection exception list.                          |
|            | antenna            | Sets the radio antenna power.   |
|            | power              | Defines the radio antenna power transmit level.   |
|            | rates              | Sets the supported radio transmit rates.  |
|            | beacon             | Sets the beacon interval used by the radio.   |
|            | dtim               | Defines the DTIM interval (by index) used by the radio.                                 |
|            | rts                | Defines the RTS Threshold value for the radio.  |
|            | range              | Sets the radio's extended range (from 0-50 miles)                                       |
|            | qos                | Defines the cwmin, cwmax, aifsn and txops levels for the QoS policy used for the radio. |
|            | qbss-beacon        | Sets the QBSS Channel Util Beacon Interval in kilo-usec (10 - 200).                     |
|            | qbss-mode          | Enables/disables the QBSS load element.   |

### **Example**

```
admin(network.wireless.radio.802-11a)>
```

```
admin(network.wireless.radio.802-11a)>set placement indoor
admin(network.wireless.radio.802-11a)>set ch-mode user
admin(network.wireless.radio.802-11a)>set channel 1
admin(network.wireless.radio.802-11a)>set acs-exception-list 44 153 161
admin(network.wireless.radio.802-11a)>set antenna full
admin(network.wireless.radio.802-11a)>set power 4
admin(network.wireless.radio.802-11a)>set rates
admin(network.wireless.radio.802-11a)>set beacon 100
admin(network.wireless.radio.802-11a)>set dtim 1 10
admin(network.wireless.radio.802-11a)>set rts 2341
admin(network.wireless.radio.802-11a)>set qos cwmin 125
admin(network.wireless.radio.802-11a)>set qos cwmax 255
admin(network.wireless.radio.802-11a)>set qos aifsn 7
admin(network.wireless.radio.802-11a)>set qos txops 0
admin(network.wireless.radio.802-11a)>set qbss-beacon 110
admin(network.wireless.radio.802-11a)>set qbss-mode enable
```

**admin(network.wireless.radio.802-11a)> advanced**

Navigates to the advanced submenu for the 802-11a radio. The items available under this command include:

**Syntax**

|             |   |
|-------------|---|
| <b>show</b> | Displays advanced radio settings for the 802-11a radio. |
| <b>set</b>  | Defines advanced parameters for the 802-11a radio.      |
| <b>..</b>   | Goes to the parent menu.                                |
| <b>/</b>    | Goes to the root menu.                                  |
| <b>save</b> | Saves the configuration to system flash.                |
| <b>quit</b> | Quits the CLI.  |

## admin(network.wireless.radio.802-11a.advanced)> show

Displays the BSSID to WLAN mapping for the 802.11a radio.

### Syntax

|             |          |   |
|-------------|----------|---|
| <b>show</b> | advanced | Displays advanced settings for the 802.11a radio. |
|             | wlan     | Displays WLAN summary list for 802.11a radio.     |

### Example

```
admin(network.wireless.radio.802-11a.advanced)>show advanced
```

| WLAN   | BSS ID | BC/MC Cipher | Status | Message             |
|--------|--------|--------------|--------|---------------------|
| Lobby  | 1      | Open         | good   | configuration is ok |
| HR     | 2      | Open         | good   | configuration is ok |
| Office | 3      | Open         | good   | configuration is ok |

| BSSID | Primary WLAN |
|-------|--------------|
| 1     | Lobby        |
| 2     | HR           |
| 3     | Office       |

```
admin(network.wireless.radio.802-11bg.advanced)>show wlan
```

```
WLAN 1:
WLAN name      : WLAN1
ESS ID         : 101
Radio          : 11a, 11b/g
VLAN           : <none>
Security Policy : Default
QoS Policy     : Default
```

**admin(network.wireless.radio.802-11a.advanced)> set**

Defines advanced parameters for the target 802..11a radio.

**Syntax**

|            |      |             |             |  |
|------------|------|-------------|-------------|--|
| <b>set</b> | wlan | <wlan-name> | <bssid>     | Defines advanced WLAN to BSSID mapping for the target radio. |
|            | bss  | <bss-id>    | <wlan name> | Sets the BSSID to primary WLAN definition.                   |

**Example**

```
admin(network.wireless.radio.802-11a.advanced)>set wlan demoroom 1
admin(network.wireless.radio.802-11a.advanced)>set bss 1 demoroom
```

## Network Quality of Service (QoS) Commands

### **admin(network.wireless)> qos**

Displays the access point *Quality of Service* (QoS) submenu. The items available under this command include:

|               |   |
|---------------|---|
| <b>show</b>   | Displays access point QoS policy information. |
| <b>create</b> | Defines the parameters of the QoS policy.     |
| <b>edit</b>   | Edits the settings of an existing QoS policy. |
| <b>delete</b> | Removes an existing QoS policy.               |
| <b>..</b>     | Goes to the parent menu.                      |
| <b>/</b>      | Goes to the root menu.                        |
| <b>save</b>   | Saves the configuration to system flash.      |
| <b>quit</b>   | Quits the CLI.                                |

**admin(network.wireless.qos)> show**

Displays the access point's current QoS policy by summary or individual policy.

**Syntax**

**show**           summary           Displays all existing QoS policies that have been defined.  
                   policy           <index>       Displays the configuration for the requested QoS policy.

**Example**

```
admin(network.wireless.qos)>show summary
```

```
-----
QoS Policy Name      Associated WLANs
-----
1 Default            WLAN1, mudskipper
2 IP Phones          Audio Dept
3 Video              Vidio Dept
```

```
admin(network.wireless.qos)>show policy 1
```

```
Policy Name          IP Phones
Support Legacy Voice Mode  disable
Multicast (Mask) Address 1  01005E000000
Multicast (Mask) Address 2  09000E000000
WMM QOS Mode         disable
```



**admin(network.wireless.qos)> create**

Navigates to a menu used to define an access point’s QoS policy.

**Syntax**

|                   |           |                   |   |
|-------------------|-----------|-------------------|---|
| <b>show</b>       |           |                   | Displays QoS policy parameters.   |
| <b>set</b>        | qos-name  | <index>           | Sets the QoS name for the specified index entry.  |
|                   | vop       | <index>           | Enables or disables support (by index) for legacy VOIP devices.   |
|                   | mcast     | <mac>             | Defines primary and secondary Multicast MAC address.  |
|                   | wmm-qos   | <index>           | Enables or disables the QoS policy index specified.   |
|                   | param-set | <set-name>        | Defines the data type used with the qos policy and mesh network. When set to a value other than manual, editing the access category values is not necessary. Options include; 11g-default, 11b-default, 11g-wifi, 11b-wifi, 11g-voice, 11b-voice or manual for advanced users). |
|                   | cwmin     | <access category> | <index> Defines Minimum Contention Window (CW-Min) for specified access category and index.   |
|                   | cwmax     | <access category> | <index> Defines Maximum Contention Window (CW-Max) for specified access category and index.   |
|                   | aifsn     | <access category> | <index> Sets Arbitrary Inter-Frame Space Number (AIFSN) for specified access category and index.  |
|                   | txops     | <access category> | <index> Configures Opportunity to Transmit Time (TXOPs Time) for specified access category and index.   |
| <b>add-policy</b> | default   | <index>           | Defines CWMIN, CWMAX, AIFSN and TXOPs default values.   |
| <b>..</b>         |           |                   | Completes the policy edit and exits the session.<br>Cancels the changes and exits.  |

**admin(network.wireless.qos)> edit**

Navigates to menu used to edit the properties of an existing QoS policy.

**Syntax**

|               |           |                           |   |
|---------------|-----------|---------------------------|---|
| <b>show</b>   |           |                           | Displays QoS policy parameters.   |
| <b>set</b>    | qos-name  | <index>                   | Sets the QoS name for the specified index entry.  |
|               | vop       | <index>                   | Enables or disables support (by index) for legacy VOIP devices.   |
|               | mcast     | <mac>                     | Defines primary and secondary Multicast MAC address.  |
|               | wmm-qos   | <index>                   | Enables or disables the QoS policy index specified.   |
|               | param-set | <set-name>                | Defines the data type used with the qos policy and mesh network. When set to a value other than manual, editing the access category values is not necessary. Options include; 11g-default, 11b-default, 11g-wifi, 11b-wifi, 11g-voice, 11b-voice or manual for advanced users). |
|               | cwmin     | <access category> <index> | Defines Minimum Contention Window (CW-Min) for specified access category and index.   |
|               | cwmax     | <access category> <index> | Defines Maximum Contention Window (CW-Max) for specified access category and index.   |
|               | aifsn     | <access category> <index> | Sets Arbitrary Inter-Frame Space Number (AIFSN) for specified access category and index.  |
|               | txops     | <access category> <index> | Configures Opportunity to Transmit Time (TXOPs Time) for specified access category and index.   |
|               | default   | <index>                   | Defines CWMIN, CWMAX, AIFSN and TXOPs default values.   |
| <b>change</b> |           |                           | Completes the policy edit and exits the session.  |
| <b>..</b>     |           |                           | Cancels the changes and exits.  |

## **admin(network.wireless.qos)> delete**

Removes a QoS policy.

### **Syntax**

**delete**                    <qos-name> <all> Deletes the specified QoS polciy index, or all of the policies (except default policy).

## Network Bandwidth Management Commands

### **admin(network.wireless)> bandwidth**

Displays the access point Bandwidth Management submenu. The items available under this command include:

|             |  |
|-------------|--|
| <b>show</b> | Displays Bandwidth Management information for how data is processed by the access point. |
| <b>set</b>  | Defines Bandwidth Management parameters for the access point.                            |
| <b>..</b>   | Goes to the parent menu.   |
| <b>/</b>    | Goes to the root menu.   |
| <b>save</b> | Saves the configuration to system flash.   |
| <b>quit</b> | Quits the CLI.   |

## **admin(network.wireless.bandwidth)> show**

Displays the current Bandwidth Management configuration summary or for defined WLANs as well as how they are weighted.

### **Syntax**

|             |                        |  |
|-------------|------------------------|--|
| <b>show</b> | <b>&lt;summary&gt;</b> | Displays the current Bandwidth Management configuration summary or for defined |
|             | <b>&lt;wlan&gt;</b>    | WLANs as well as how they are weighted.  |

### **Example**

```
admin(network.wireless.bandwidth)>show summary
```

|                        |                      |
|------------------------|----------------------|
| Bandwidth Share Mode 1 | : First In First Out |
| Bandwidth Share Mode 2 | : First In First Out |

**admin(network.wireless.bandwidth)> set**

Defines the access point Bandwidth Management configuration.

**Syntax**

|            |        |           |  |
|------------|--------|-----------|--|
| <b>set</b> | mode   | <bw-mode> | Defines bandwidth share mode of First In First Out <fifo>, Round Robin <rr> or Weighted Round Robin <wrr>                                |
|            | weight | <num>     | Assigns a bandwidth share allocation for the WLAN <index 1-16 > when Weighted Round Robin <wrr> is selected. The weighting is from 1-10. |

## Network Rogue-AP Commands

### **admin(network.wireless)> rogue-ap**

Navigates to the Rogue AP submenu. The items available under this command include:

|                     |   |
|---------------------|---|
| <b>show</b>         | Displays the current access point Rogue AP detection configuration. |
| <b>set</b>          | Defines the Rogue AP detection method.                              |
| <b>mu-scan</b>      | Goes to the Rogue AP scan submenu.                                  |
| <b>allowed-list</b> | Goes to the Rogue AP Allowed List submenu.                          |
| <b>active-list</b>  | Goes the Rogue AP Active List submenu.                              |
| <b>rogue-list</b>   | Goes the Rogue AP List submenu.                                     |
| <b>..</b>           | Goes to the parent menu.  |
| <b>/</b>            | Goes to the root menu.  |
| <b>save</b>         | Saves the configuration to system flash.                            |
| <b>quit</b>         | Quits the CLI.  |

**admin(network.wireless.rogue-ap)> show**

Displays the current access point Rogue AP detection configuration.

**Syntax**

**show** Displays the current access point Rogue AP detection configuration.

**Example**

```
admin(network.wireless.rogue-ap)>show
```

```
MU Scan                               : disable
MU Scan Interval                       : 60 minutes
On-Channel                             : disable
Detector Radio Scan                    : enable

Auto Authorize Extreme Networks APs    : disable

Approved APs age out                   : 0 minutes
Rogue APs age out                       : 0 minutes
```



## admin(network.wireless.rogue-ap)> set

Defines the access point ACL rogue AP method.

### Syntax

|            |               |           |   |
|------------|---------------|-----------|---|
| <b>set</b> | mu-scan       | <mode>    | Enables or disables to permit Clients to scan for rogue APs.  |
|            | interval      | <minutes> | Define an interval for associated Clients to beacon in attempting to locate rogue APs. Value not available unless mu-scan is enabled. |
|            | on-channel    | <mode>    | Enables or disables on-channel detection.   |
|            | detector-scan | <mode>    | Enables or disables AP detector scan (dual-radio model only).   |
|            | ABG-scan      | <mode>    | Enables or disables A/BG Detector Scan Mode.  |
|            | extreme       | <mode>    | Enables or disables the Authorize Any AP with a Extreme Networks MAC address option.  |
|            | networks-ap   |           |   |
|            | applst-ageout | <minutes> | Sets the approved AP age out time.  |
|            | roglst-ageout | <minutes> | Sets the rogue AP age out time.   |

### Example

```
admin(network.wireless.rogue-ap)>
```

```
admin(network.wireless.rogue-ap)>set mu-scan enable
admin(network.wireless.rogue-ap)>set interval 10
admin(network.wireless.rogue-ap)>set on-channel disable
admin(network.wireless.rogue-ap)>set detector-scan disable
admin(network.wireless.rogue-ap)>set ABG-scan disable
admin(network.wireless.rogue-ap)>set extreme networks-ap enable
admin(network.wireless.rogue-ap)>set applst-ageout 10
admin(network.wireless.rogue-ap)>set roglst-ageout 10
```

```
admin(network.wireless.rogue-ap)>show
```

```
MU Scan                               : enable
MU Scan Interval                       : 10 minutes
On Channel                             : disable
Detector Radio Scan                    : disable

Auto Authorize Extreme Networks APs    : enable

Approved AP age out                    : 10 minutes
Rogue AP age out                       : 10 minutes
```

**admin(network.wireless.rogue-ap)> mu-scan**

Navigates to the Rogue-AP mu-scan submenu.

**Syntax**

|              |   |
|--------------|---|
| <b>add</b>   | Add all or just one scan result to Allowed AP list. |
| <b>show</b>  | Displays all APs located by the Client scan.        |
| <b>start</b> | Initiates scan immediately by the Client.           |
| <b>..</b>    | Goes to the parent menu.                            |
| <b>/</b>     | Goes to the root menu.                              |
| <b>save</b>  | Saves the configuration to system flash.            |
| <b>quit</b>  | Quits the CLI.                                      |

## **admin(network.wireless.rogue-ap.mu-scan)> start**

Initiates an MU scan for a user provided MAC address.

### **Syntax**

|              |          |   |
|--------------|----------|---|
| <b>start</b> | <mu-mac> | Initiates Client scan from user provided MAC address. |
|--------------|----------|---|

**admin(network.wireless.rogue-ap.mu-scan)> show**

Displays the results of an MU scan.

### **Syntax**

**show**                Displays all APs located by the Client scan.

## **admin(network.wireless.rogue-ap)> allowed-list**

Navigates to the Rogue-AP allowed-list submenu.

|               |  |
|---------------|--|
| <b>show</b>   | Displays the rogue AP allowed list                     |
| <b>add</b>    | Adds an AP MAC address and ESSID to the allowed list.  |
| <b>delete</b> | Deletes an entry or all entries from the allowed list. |
| <b>..</b>     | Goes to the parent menu.                               |
| <b>/</b>      | Goes to the root menu.                                 |
| <b>save</b>   | Saves the configuration to system flash.               |
| <b>quit</b>   | Quits the CLI.   |

**admin(network.wireless.rogue-ap.allowed-list)> show**

Displays the Rogue AP allowed List.

**Syntax**

**show**                Displays the Rogue AP allowed List.

**Example**

```
admin(network.wireless.rogue-ap.allowed-list)>show
```

| Allowed AP List |                   |           |
|-----------------|-------------------|-----------|
| index           | ap mac            | essid     |
| 1               | 00:A0:F8:71:59:20 | *         |
| 2               | 00:A0:F8:33:44:55 | 101       |
| 3               | 00:A0:F8:40:20:01 | Marketing |

## **admin(network.wireless.rogue-ap.allowed-list)> add**

Adds an AP MAC address and ESSID to existing allowed list.

### **Syntax**

**add**                    <mac-addr>       Adds an AP MAC address and ESSID to existing allowed list.  
                         <ess-id>           "ffffffffffffffff" means any MAC  
                                   Use a "\*" for any ESSID.

### **Example**

```
admin(network.wireless.rogue-ap.allowed-list)>add 00A0F83161BB 103
admin(network.wireless.rogue-ap.allowed-list)>show
```

```
-----
index          ap          essid
-----
1              00:A0:F8:71:59:20      *
2              00:A0:F8:33:44:55      ffffffffffffffff
3              00:A0:F8:40:20:01      Marketing
4              00:A0:F8:31:61:BB      103
```

**admin(network.wireless.rogue-ap.allowed-list)> delete**

Deletes an AP MAC address and ESSID to existing allowed list.

**Syntax**

|               |       |  |
|---------------|-------|--|
| <b>delete</b> | <idx> | Deletes a specified AP MAC address and ESSID index (1-50) from the allowed |
|               | <all> | list. The option also exists to remove all indexes.                        |



## WIPS Commands

### **admin(network.wireless)> wips**

Navigates to the wips Locationing submenu. The items available under this command include:

|             |   |
|-------------|---|
| <b>show</b> | Displays the current WLAN Intrusion Prevention configuration. |
| <b>set</b>  | Sets WLAN Intrusion Prevention parameters.                    |
| <b>..</b>   | Goes to the parent menu.                                      |
| <b>/</b>    | Goes to the root menu.  |
| <b>save</b> | Saves the configuration to system flash.                      |
| <b>quit</b> | Quits the CLI.  |

**admin(network.wireless.wips)> show**

Shows the WLAN Intrusion Prevention configuration.

**Syntax**

**show**      Shows the WLAN Intrusion Prevention configuration.

**Example**

```
admin(network.wireless.wips)>show
```

```
WIPS Server #1
  IP Address           : 192.168.0.21
```

```
WIPS Server #2
  IP Address           : 10.10.1.1
```

```
admin(network.wireless.wips)>
```

## **admin(network.wireless.wips)> set**

Sets the WLAN Intrusion Prevention configuration.

### **Syntax**

**set**            <idx 1 and 2> <ip> Defines the WLAN Intrusion Prevention Server IP Address for (server IPs 1 and 2)

### **Example**

```
admin(network.wireless.wips)>set server 1 192.168.0.21
admin(network.wireless.wips>
```

## Network MU Locationing Commands

### **admin(network.wireless)> mu-locationing**

Navigates to the Client Locationing submenu. The items available under this command include:

|             |  |
|-------------|--|
| <b>show</b> | Displays the current Client Locationing configuration. |
| <b>set</b>  | Defines Client Locationing parameters.                 |
| <b>..</b>   | Goes to the parent menu.                               |
| <b>/</b>    | Goes to the root menu.                                 |
| <b>save</b> | Saves the configuration to system flash.               |
| <b>quit</b> | Quits the CLI.   |

## **admin(network.wireless.mu-locationing)> show**

Displays the MU probe table configuration

### **Syntax**

**show**        Displays the Client probe table configuration.

### **Example**

```
admin(network.wireless.mu-locationing)>show
```

```
MU Probe Table Mode           : disable
MU Probe Table Size           : 200
```

```
admin(network.wireless.mu-locationing)>
```

**admin(network.wireless.mu-locationing)> set**

Defines the MU probe table configuration used for locating MUs.

**Syntax**

|            |      |  |
|------------|------|--|
| <b>set</b> |      | Defines the probe table configuration.                                   |
|            | mode | Enables/disables a probe scan for the purposes of MU locationing.        |
|            | size | Defines the number of Clients in the table (the maximum allowed is 200). |

**Example**

```
admin(network.wireless.mu-locationing)>set
```

```
admin(network.wireless.mu-locationing)>set mode enable
```

```
admin(network.wireless.mu-locationing)>set size 200
```

```
admin(network.wireless.mu-locationing)>
```

## Network Reliable Multicast Commands

### **admin(network.wireless)> reliable-multicast**

Navigates to the Reliable Multicast submenu. The items available under this command include:

|               |   |
|---------------|---|
| <b>add</b>    | Adds a multicast streaming address for Reliable Multicast.  |
| <b>delete</b> | Removes multicast streaming address for Reliable Multicast. |
| <b>show</b>   | Displays the current Reliable Multicast configuration.      |
| <b>set</b>    | Defines the Reliable Multicast configuration information.   |
| <b>..</b>     | Goes to the parent menu.                                    |
| <b>/</b>      | Goes to the root menu.                                      |
| <b>save</b>   | Saves the configuration to system flash.                    |
| <b>quit</b>   | Quits the CLI.  |

**admin(network.wireless.reliable-multicast)> add**

Adds a multicast address for Reliable Multicast feature.

**Syntax :**

**add**    multicast-group            Adds a multicast group for Reliable Multicast feature  
          <IPv4 Multicast group> The multicast group to be added. The value for this parameter is an IP address in the range of 244.0.0.0 to 239.255.255.255.

**Example**

```
admin(network.wireless.reliable-multicast)>add multicast-group 224.0.1.10
admin(network.wireless.reliable-multicast)>show config
Reliable Multicast mode           : disable
Action on original M/C packet    : Drop
Reliable Multicast WLAN          : Reliable_Multicast
Max Streams to be serviced       : 12
Standalone mode                  : disable
IGMP Query interval              : 60
IGMP Query version               : IGMPv3
Reliable Multicast Groups        : 224.0.1.10
```



## **admin(network.wireless.reliable-multicast)> delete**

Removes multicast address or addresses for Reliable Multicast feature.

### **Syntax :**

|               |                        |  |
|---------------|------------------------|--|
| <b>delete</b> | multicast-group        | Removes a multicast group for Reliable Multicast feature. <IPv4 Multicast Group> is the multicast group to be removed. The value for this parameter is an IP address in the range of 244.0.0.0 to 239.255.255.255. |
|               | <IPv4 Multicast Group> |  |
|               | all                    | Removes all multicast groups registered for Reliable Multicast.  |

### **Example**

```
admin(network.wireless.reliable-multicast)>show config
```

```
Reliable Multicast mode      : disable
Action on original M/C packet : Drop
Reliable Multicast WLAN      : Reliable_Multicast
Max Streams to be serviced   : 12
Standalone mode              : disable
IGMP Query interval          : 60
IGMP Query version           : IGMPv3
Reliable Multicast Groups    : 224.0.1.10
                             : 224.0.1.20
                             : 224.0.2.10
                             : 224.0.2.20
                             : 224.0.3.10
                             : 224.0.3.20
```

```
admin(network.wireless.reliable-multicast)>delete multicast-group 224.0.2.20
Multicast groups successfully removed.
```

```
admin(network.wireless.reliable-multicast)>show config
```

```
Reliable Multicast mode      : disable
Action on original M/C packet : Drop
Reliable Multicast WLAN      : Reliable_Multicast
Max Streams to be serviced   : 12
Standalone mode              : disable
IGMP Query interval          : 60
IGMP Query version           : IGMPv3
Reliable Multicast Groups    : 224.0.1.10
                             : 224.0.1.20
                             : 224.0.2.10
                             : 224.0.3.10
                             : 224.0.3.20
```

```
admin(network.wireless.reliable-multicast)>
admin(network.wireless.reliable-multicast)>delete all
All Multicast groups successfully removed.
```

```
admin(network.wireless.reliable-multicast)>show config
```

```
Reliable Multicast mode      : disable
Action on original M/C packet : Drop
Reliable Multicast WLAN      : Reliable_Multicast
Max Streams to be serviced   : 12
Standalone mode              : disable
IGMP Query interval          : 60
IGMP Query version           : IGMPv3
Reliable Multicast Groups    :
admin(network.wireless.reliable-multicast)>
```

**admin(network.wireless.reliable-multicast)> set**

Sets the different Reliable Multicast configuration settings.

**Syntax :**

|            |                 |           |  |
|------------|-----------------|-----------|--|
| <b>set</b> | mode            | <mode>    | Enables or disables the Reliable Multicast feature.  |
|            | stream-limit    | <count>   | Sets the number of Multicast streams supported by Reliable Multicast. Enter a value in the range <i>1</i> and <i>32</i> . The default value is <i>12</i> .           |
|            | query-interval  | <seconds> | Sets the IGMP query interval in seconds. Enter a value in the range <i>30</i> and <i>300</i> .   |
|            | query-version   | <version> | Sets the IGMP version to use. Use <i>IGMPv1</i> or <i>IGMPv2</i> or <i>IGMPv3</i> .  |
|            | standalone-mode | <mode>    | Enables or disable stand alone mode for Reliable Multicast. In this mode, the AP assumes the role of a IGMP querier in the absence of an IGMP router in the network. |
|            | wlan            | <idx>     | Enables the WLAN with the ESS ID <idx> for Reliable Multicast.   |
|            | tx-multicast    | <mode>    | Enables the re-transmission of the received Multicast packet. If disabled, the received multicast packet is dropped after converting it to a unicast packet.         |

**Example**

```
admin(network.wireless.reliable-multicast)>set mode enable
admin(network.wireless.reliable-multicast)>set stream-limit 10
admin(network.wireless.reliable-multicast)>set query-interval 55
admin(network.wireless.reliable-multicast)>set query-version IGMPv2
admin(network.wireless.reliable-multicast)>set standalone-mode disable
admin(network.wireless.reliable-multicast)>set wlan Reliable_Multicast
admin(network.wireless.reliable-multicast)>set tx-multicast disable
admin(network.wireless.reliable-multicast)>show config
Reliable Multicast mode           : enable
Action on original M/C packet    : Drop
Reliable Multicast WLAN          : Reliable_Multicast
Max Streams to be serviced       : 10
Standalone mode                  : disable
IGMP Query interval              : 55
IGMP Query version               : IGMPv2
Reliable Multicast Groups        :
admin(network.wireless.reliable-multicast)>
```

## admin(network.wireless.reliable-multicast)> show

Displays the configuration information for the Reliable Multicast feature. Also displays the MUs that are subscribed for Reliable Multicast transmission.

### Syntax :

|             |              |  |
|-------------|--------------|--|
| <b>show</b> | config       | Displays the current configuration for the Reliable Multicast feature.   |
|             | mobile-units | Displays the MUs that are subscribed to Reliable Multicast transmission. |

### Example

```
admin(network.wireless.reliable-multicast)>show config
Reliable Multicast mode           : enable
Action on original M/C packet     : Drop
Reliable Multicast WLAN           : Reliable_Multicast
Max Streams to be serviced        : 10
Standalone mode                   : disable
IGMP Query interval               : 55
IGMP Query version                : IGMPv2
Reliable Multicast Groups         :
```

```
admin(network.wireless.reliable-multicast)>show mobile-units
```

```
-----
239.1.1.1      239.1.1.5  239.1.1.6
239.1.1.7
-----
```

```
00:40:96:b3:e8:8100:40:96:b3:e8:8100:40:96:b3:e8:81
00:40:96:a8:4e:6700:40:96:a8:4e:6700:40:96:a8:4e:67
```

## Network DOT 11i Retry Commands

### **admin(network.wireless)> dot11i-retry**

Navigates to the 11i retry command submenu:

|             |   |
|-------------|---|
| <b>show</b> | Displays the current dot11i retry configuration.    |
| <b>set</b>  | Defines the dot11i retry configuration information. |
| <b>..</b>   | Goes to the parent menu.                            |
| <b>/</b>    | Goes to the root menu.                              |
| <b>save</b> | Saves the configuration to system flash.            |
| <b>quit</b> | Quits the CLI.                                      |

## **admin(network.wireless.dot11i-retry)> show**

Displays the configuration information for the dot11i retry feature.

### **Syntax :**

**show** <idx> Displays the retry configuration for the WLAN specified by the index <idx>.

### **Example**

```
admin(network.wireless.dot11i-retry)>show 1
```

```
handshake timeout in milliseconds: 2000
handshake retry count             : 3
```

```
admin(network.wireless.dot11i-retry)>
```

**admin(network.wireless.dot11i-retry)> set**

Sets the configuration parameters for the dot11i retry feature.

**Syntax :**

|            |                       |       |                         |   |
|------------|-----------------------|-------|-------------------------|---|
| <b>set</b> | handshake-timeout     | <idx> | <handshake-timeout>     | Sets the handshake retry timeout value for the WLAN specified by the index <idx> to the duration in ms specified by the <handshake-timeout> (value between <i>100-2000</i> ) parameter. |
|            | handshake-retry-count | <idx> | <handshake-retry-count> | Sets the handshake retry count for the WLAN specified by the index <idx> to <handshake-retry-count> (value between <i>1</i> and <i>10</i> )   |

**Example**

```
admin(network.wireless.dot11i-retry)>set handshake-timeout 1 200
admin(network.wireless.dot11i-retry)>set handshake-retry-count 1 6
admin(network.wireless.dot11i-retry)>show 1
```

```
handshake timeout in milliseconds: 200
handshake retry count           : 6
```

```
admin(network.wireless.dot11i-retry)>
```

## Network Firewall Commands

### **admin(network)> firewall**

Navigates to the access point firewall submenu. The items available under this command include:

|                 |  |
|-----------------|--|
| <b>show</b>     | Displays the access point's current firewall configuration.          |
| <b>set</b>      | Defines the access point's firewall parameters.                      |
| <b>access</b>   | Enables/disables firewall permissions through the LAN and WAN ports. |
| <b>advanced</b> | Displays interoperability rules between the LAN and WAN ports.       |
| <b>..</b>       | Goes to the parent menu.   |
| <b>/</b>        | Goes to the root menu.   |
| <b>save</b>     | Saves the configuration to system flash.                             |
| <b>quit</b>     | Quits the CLI.   |

**admin(network.firewall)> show**

Displays the access point firewall parameters.

**Syntax**

**show**                      Shows all access point's firewall settings.

**Example**

```
admin(network.firewall)>show
```

```
Firewall Status           : disable
NAT Timeout               : 10 minutes
```

Configurable Firewall Filters:

```
ftp bounce attack filter  : enable
syn flood attack filter   : enable
unaligned ip timestamp filter : enable
source routing attack filter : enable
winnuke attack filter     : enable
seq num prediction attack filter : enable
mime flood attack filter  : enable
max mime header length    : 8192 bytes
max mime headers          : 16 headers
```



## admin(network.firewall)> set

Defines the access point firewall parameters.

### Syntax

|            |             |            |   |
|------------|-------------|------------|---|
| <b>set</b> | mode        | <mode>     | Enables or disables the firewall.   |
|            | nat-timeout | <interval> | Defines the NAT timeout value.  |
|            | syn         | <mode>     | Enables or disables SYN flood attack check.   |
|            | src         | <mode>     | Enables or disables source routing check.   |
|            | win         | <mode>     | Enables or disables Winnuke attack check.   |
|            | ftp         | <mode>     | Enables or disables FTP bounce attack check.  |
|            | ip          | <mode>     | Enables or disables IP unaligned timestamp check.   |
|            | seq         | <mode>     | Enables or disables sequence number prediction check.   |
|            | mime        | filter     | Enables or disables MIME flood attack check.  |
|            | len         | <length>   | Sets the max header length in bytes as specified by <length> (with value in range <i>256 - 34463</i> ). |
|            | hdr         | <count>    | Sets the max number of headers as specified in <count> (with value in range <i>12 - 34463</i> ).        |

### Example

```
admin(network.firewall)>set mode enable
admin(network.firewall)>set ftp enable
admin(network.firewall)>set ip enable
admin(network.firewall)>set seq enable
admin(network.firewall)>set src enable
admin(network.firewall)>set syn enable
admin(network.firewall)>set win enable
admin(network.firewall)>show
```

```
Firewall Status           : enable
Override LAN to WAN Access : disable
```

#### Configurable Firewall Filters

```
ftp bounce attack filter   : enable
syn flood attack filter    : enable
unaligned ip timestamp filter : enable
source routing attack filter : enable
winnuke attack filter      : enable
seq num prediction attack filter : enable
mime flood attack filter   : enable
max mime header length     : 8192
max mime headers           : 16
```

**admin(network.firewall)> access**

Enables or disables firewall permissions through LAN to WAN ports.

**Syntax**

|               |   |
|---------------|---|
| <b>show</b>   | Displays LAN to WAN access rules.           |
| <b>set</b>    | Sets LAN to WAN access rules.               |
| <b>add</b>    | Adds LAN to WAN exception rules.            |
| <b>delete</b> | Deletes LAN to WAN access exception rules.  |
| <b>list</b>   | Displays LAN to WAN access exception rules. |
| <b>..</b>     | Goes to parent menu                         |
| <b>/</b>      | Goes to root menu.                          |
| <b>save</b>   | Saves configuration to system flash.        |
| <b>quit</b>   | Quits and exits the CLI session.            |

**Example**

```
admin(network.firewall.lan-wan-access)>list
```

| index | from | to  | name   | prot | start port | end port |
|-------|------|-----|--------|------|------------|----------|
| 1     | lan  | wan | HTTP   | tcp  | 80         | 80       |
| 2     | lan  | wan | abc    | udp  | 0          | 0        |
| 3     | lan  | wan | 123456 | ah   | 1440       | 2048     |
| 4     | lan  | wan | 654321 | tcp  | 2048       | 2048     |
| 5     | lan  | wan | abc    | ah   | 100        | 1000     |

## admin(network.firewall)> advanced

Displays whether an access point firewall rule is intended for inbound traffic to an interface or outbound traffic from that interface..

### Syntax

|                 |  |
|-----------------|--|
| <b>show</b>     | Shows advanced subnet access parameters.     |
| <b>set</b>      | Sets advanced subnet access parameters.      |
| <b>import</b>   | Imports rules from subnet access.            |
| <b>inbound</b>  | Goes to the Inbound Firewall Rules submenu.  |
| <b>outbound</b> | Goes to the Outbound Firewall Rules submenu. |
| <b>..</b>       | Goes to the parent menu.                     |
| <b>/</b>        | Goes to the root menu.                       |
| <b>save</b>     | Saves the configuration to flash memory.     |
| <b>quit</b>     | Quits and exits the CLI session.             |

### Example

```
admin(network.firewall.adv-lan-access)>inbound
admin(network.firewall.adv-lan-access.inb)>list
```

| Idx | SCR IP-Netmask            | Dst IP-Netmask             | TP  | SPorts      | DPorts      | Rev       | NAT        | Action |
|-----|---------------------------|----------------------------|-----|-------------|-------------|-----------|------------|--------|
| 1   | 1.2.3.4<br>255.0.0.0      | 2.2.2.2<br>255.0.0.0       | all | 1:<br>65535 | 1:<br>65535 | 0.0.0.0   |            | deny   |
| 2   | 33.3.0.0<br>255.255.255.0 | 10.10.1.1<br>255.255.255.0 | tcp | 1:<br>65535 | 1:<br>65535 | 11.11.1.0 | nat port 0 | allow  |

## Network Router Commands

### **admin(network)> router**

Navigates to the router submenu. The items available under this command are:

|               |  |
|---------------|--|
| <b>show</b>   | Displays the existing access point router configuration. |
| <b>set</b>    | Sets the RIP parameters.                                 |
| <b>add</b>    | Adds user-defined routes.                                |
| <b>delete</b> | Deletes user-defined routes.                             |
| <b>list</b>   | Lists user-defined routes.                               |
| <b>..</b>     | Goes to the parent menu.                                 |
| <b>/</b>      | Goes to the root menu.                                   |
| <b>save</b>   | Saves the configuration to system flash.                 |
| <b>quit</b>   | Quits the CLI.   |

## **admin(network.router)> show**

Shows the access point route table.

### **Syntax**

**show**       Shows the access point route table.

### **Example**

```
admin(network.router)>show routes
```

| index | destination  | netmask       | gateway      | interface | metric |
|-------|--------------|---------------|--------------|-----------|--------|
| 1     | 192.168.2.0  | 255.255.255.0 | 0.0.0.0      | lan1      | 0      |
| 2     | 192.168.1.0  | 255.255.255.0 | 0.0.0.0      | lan2      | 0      |
| 3     | 192.168.0.0  | 255.255.255.0 | 0.0.0.0      | lan1      | 0      |
| 4     | 192.168.24.0 | 255.255.255.0 | 0.0.0.0      | wan       | 0      |
| 5     | 157.235.19.5 | 255.255.255.0 | 192.168.24.1 | wan       | 1      |

Default gateway Interface : lan1

**admin(network.router)> set**

Sets access point route table entries.

**Syntax**

|            |           |  |
|------------|-----------|--|
| <b>set</b> | auth      | Sets the RIP authentication type.            |
|            | dir       | Sets RIP direction.                          |
|            | id        | Sets MD5 authentication ID.                  |
|            | key       | Sets MD5 authentication key.                 |
|            | passwd    | Sets the password for simple authentication. |
|            | type      | Defines the RIP type.                        |
|            | dgw-iface | Sets the default gateway interface.          |

## **admin(network.router)> add**

Adds user-defined routes.

### **Syntax**

**add** <dest> <netmask> <gw> <iface> <metric> Adds a route with destination IP address <dest>, IP netmask <netmask>, destination gateway IP address <gw>, interface LAN1, LAN2 or WAN <iface>, and metric set to <metric> (1-65536).

### **Example**

```
admin(network.router)>add 192.168.3.0 255.255.255.0 192.168.2.1 LAN1 1
```

```
admin(network.router)>list
```

| index | destination | netmask       | gateway     | interface | metric |
|-------|-------------|---------------|-------------|-----------|--------|
| 1     | 192.168.3.0 | 255.255.255.0 | 192.168.2.1 | lan1      | 1      |

**admin(network.router)> delete**

Deletes user-defined routes.

**Syntax**

**delete**      <idx>      Deletes the user-defined route <idx> (1-20) from list.  
                  all            Deletes all user-defined routes.

**Example**

```
admin(network.router)>list
```

| index | destination | netmask       | gateway     | interface | metric |
|-------|-------------|---------------|-------------|-----------|--------|
| 1     | 192.168.2.0 | 255.255.255.0 | 192.168.0.1 | lan1      | 1      |
| 2     | 192.168.1.0 | 255.255.255.0 | 0.0.0.0     | lan2      | 0      |
| 3     | 192.168.0.0 | 255.255.255.0 | 0.0.0.0     | lan2      | 0      |

```
admin(network.router)>delete 2
```

```
admin(network.router)>list
```

| index | destination | netmask       | gateway | interface | metric |
|-------|-------------|---------------|---------|-----------|--------|
| 1     | 192.168.2.0 | 255.255.255.0 | 0.0.0.0 | lan1      | 0      |
| 2     | 192.168.0.0 | 255.255.255.0 | 0.0.0.0 | lan1      | 0      |

```
admin(network.router)>
```



## **admin(network.router)> list**

Lists user-defined routes.

### **Syntax**

**list**                Displays a list of user-defined routes.

### **Example**

```
admin(network.router)>list
```

| index | destination | netmask       | gateway     | interface | metric |
|-------|-------------|---------------|-------------|-----------|--------|
| 1     | 192.168.2.0 | 255.255.255.0 | 192.168.0.1 | lan1      | 1      |
| 2     | 192.168.1.0 | 255.255.255.0 | 0.0.0.0     | lan2      | 0      |
| 3     | 192.168.0.0 | 255.255.255.0 | 0.0.0.0     | lan1      | 0      |

## System Commands

### **admin>system**

Navigates to the System submenu. The items available under this command are shown below.

|                  |   |
|------------------|---|
| <b>restart</b>   | Restarts the access point.  |
| <b>show</b>      | Shows access point system parameter settings.   |
| <b>set</b>       | Defines access point system parameter settings.   |
| <b>lastpw</b>    | Displays last debug password.   |
| <b>exec</b>      | Goes to a Linux command menu.   |
| <b>arp</b>       | Displays the access point's arp table.  |
| <b>aap-setup</b> | Goes to the AP Settings submenu.  |
| <b>access</b>    | Goes to the access point access submenu where access point access methods can be enabled. |
| <b>cmgr</b>      | Goes to the Certificate Manager submenu.  |
| <b>snmp</b>      | Goes to the SNMP submenu.   |
| <b>userdb</b>    | Goes to the user database submenu.  |
| <b>radius</b>    | Goes to the Radius submenu.   |
| <b>ntp</b>       | Goes to the Network Time Protocol submenu.  |
| <b>logs</b>      | Displays the log file submenu.  |
| <b>config</b>    | Goes to the configuration file update submenu.  |
| <b>fw-update</b> | Goes to the firmware update submenu.  |
| <b>..</b>        | Goes to the parent menu.  |
| <b>/</b>         | Goes to the root menu.  |
| <b>save</b>      | Saves the configuration to system flash.  |
| <b>quit</b>      | Quits the CLI.  |

## **admin(system)> restart**

Restarts the access point access point.

### **Syntax**

**restart**                Restarts the access point.

### **Example**

```
admin(system)>restart
```

```
*****WARNING*****
** Unsaved configuration changes will be lost when the access point is reset.
** Please be sure to save changes before resetting.
*****
```

```
Are you sure you want to restart the AP35xx?? (yes/no):
```

```
AP35xx Boot Firmware Version 4.0.0.0-XXX
Copyright(c) Extreme Networks 2009. All rights reserved.
```

```
Press escape key to run boot firmware .....
```

```
Power On Self Test
```

```
testing ram           : pass
testing nor flash     : pass
testing nand flash    : pass
testing ethernet      : pass
```

**admin(system)> show**

Displays high-level system information helpful to differentiate this access point.

**Syntax**

**show** Displays access point system information.

**Example**

```
admin(system)>show
```

```
system name           : BldgC
system location        : Atlanta Field Office
admin email address    : johndoe@mycompany.com
system uptime          : 0 days 4 hours 41 minutes
```

```
AP35xx firmware version : 2.2.0.0-XXX
country code            : us
ap-mode                  : independent
serial number            : 05224520500336
```

```
admin(system)>
```

## **admin(system)> set**

Sets access point system parameters.

### **Syntax**

|            |       |         |  |
|------------|-------|---------|--|
| <b>set</b> | name  | <name>  | Sets the access point system name to <name> (1 to 59 characters). The access point does not allow intermediate space characters between characters within the system name. For example, "AP35xx sales" must be changed to "AP35xxsales" to be a valid system name. |
|            | loc   | <loc>   | Sets the access point system location to <loc> (1 to 59 characters).   |
|            | email | <email> | Sets the access point admin email address to <email> (1 to 59 characters).   |
|            | cc    | <code>  | Sets the access point country code using two-letters <code>.   |

**admin(system)> lastpw**

Displays last expired debug password.

**Example**

```
admin(system)>lastpw
```

```
AP35xx MAC Address is 00:15:70:02:7A:66
```

```
Last debug password was admin123
```

```
Current debug password used 0 times, valid 4 more time(s)
```

```
admin(system)>
```

## admin(system)> arp

Display the access point's arp table.

### Example

```
admin(system)>arp
```

| Address        | HWtype | HWaddress         | Flags Mask | Iface |
|----------------|--------|-------------------|------------|-------|
| 157.235.92.210 | ether  | 00:11:25:14:61:A8 | C          | ixp1  |
| 157.235.92.179 | ether  | 00:14:22:F3:D7:39 | C          | ixp1  |
| 157.235.92.248 | ether  | 00:11:25:B2:09:60 | C          | ixp1  |
| 157.235.92.180 | ether  | 00:0D:60:D0:06:90 | C          | ixp1  |
| 157.235.92.3   | ether  | 00:D0:2B:A0:D4:FC | C          | ixp1  |
| 157.235.92.181 | ether  | 00:15:C5:0C:19:27 | C          | ixp1  |
| 157.235.92.80  | ether  | 00:11:25:B2:0D:06 | C          | ixp1  |
| 157.235.92.95  | ether  | 00:14:22:F9:12:AD | C          | ixp1  |
| 157.235.92.161 | ether  | 00:06:5B:97:BD:6D | C          | ixp1  |
| 157.235.92.126 | ether  | 00:11:25:B2:29:64 | C          | ixp1  |

```
admin(system)>
```

## Adaptive AP Setup Commands

### **admin(system)> aap-setup**

Navigates to the Adaptive AP submenu.

|               |   |
|---------------|---|
| <b>show</b>   | Displays adopted AP information.                                  |
| <b>set</b>    | Defines the adopted AP's configuration.                           |
| <b>delete</b> | Deletes static controller address assignments.                    |
| <b>..</b>     | Goes to the parent menu.  |
| <b>/</b>      | Goes to the root menu.  |
| <b>save</b>   | Saves the current configuration to the access point system flash. |
| <b>quit</b>   | Quits the CLI and exits the current session.                      |



## admin(system.aap-setup)> show

Displays the access point's configuration once adopted by the controller.

### Syntax

**show** Displays the access point's adopted configuration.

### Example

```
admin(system.aap-setup)>show
```

```
Auto Discovery Mode           : disable
controller Interface          : lan1
controller Name                : greg
Static IP Port                : 24576
Static IP Address             :
IP Address 1                  : 0.0.0.0
IP Address 2                  : 0.0.0.0
IP Address 3                  : 0.0.0.0
IP Address 4                  : 0.0.0.0
IP Address 5                  : 0.0.0.0
IP Address 6                  : 0.0.0.0
IP Address 7                  : 0.0.0.0
IP Address 8                  : 0.0.0.0
IP Address 9                  : 0.0.0.0
IP Address 10                 : 0.0.0.0
IP Address 11                 : 0.0.0.0
IP Address 12                 : 0.0.0.0

Tunnel to controller          : disable
AC Keepalive                  : 5

Current Controller             : 157.235.22.11
AP Adoption State              : TBD
```

```
admin(system.aap-setup)>
```



### NOTE

*The access point CLI is only the only AP interface that displays the AP's adoption status and AP run state.*

**admin(system.aap-setup)> set**

Sets adopted access point's configuration.

**Syntax**

|            |                      |  |
|------------|----------------------|--|
| <b>set</b> | auto-discovery       | Sets the controller auto-discovery mode (enable/disable).        |
|            | interface            | Defines the tunnel interface.                                    |
|            | ipadr                | Defines the controller IP address used.                          |
|            | name                 | Defines the controller name for DNS lookups.                     |
|            | port                 | Sets the port.   |
|            | passphrase           | Defines the pass phrase or key for controller connection.        |
|            | tunnel-to-controller | Enables/disables the tunnel between controller and access point. |
|            | ac-keepalive         | Defines the keepalive interval.                                  |

## **admin(system.aap-setup)> delete**

Deletes static controller address assignments.

### **Syntax**

|               |       |  |
|---------------|-------|--|
| <b>delete</b> | <idx> | Deletes static controller address assignments by the selected index. |
|               | <all> | Deletes all assignments.   |

### **Example**

```
admin(system.aap-setup)>delete 1
```

```
admin(system.aap-setup)>
```

## System Access Commands

### **admin(system)> access**

Navigates to the access point's access submenu.

|             |   |
|-------------|---|
| <b>show</b> | Displays access point system access capabilities.                 |
| <b>set</b>  | Goes to the access point system access submenu.                   |
| <b>..</b>   | Goes to the parent menu.  |
| <b>/</b>    | Goes to the root menu.  |
| <b>save</b> | Saves the current configuration to the access point system flash. |
| <b>quit</b> | Quits the CLI and exits the current session.                      |

## admin(system.access)> set

Defines the permissions to access the access point applet, CLI, SNMP as well as defining their timeout values.

### Syntax

|            |                  |                            |  |
|------------|------------------|----------------------------|--|
| <b>set</b> | applet           |                            | Defines the applet HTTP/HTTPS access parameters.   |
|            | app-timeout      | <minutes>                  | Sets the applet timeout. Default is 300 Mins.  |
|            | cli              |                            | Defines CLI Telnet access parameters. Enables/disables access from lan and wan.                                  |
|            | ssh              |                            | Sets the CLI SSH access parameters.  |
|            | trusted-host     | <mode>,<br><range> <clear> | Enables/Disables global management access (snmp, http, https, telnet and ssh) for up to 8 addresses (hosts).     |
|            | auth-timout      | <seconds>                  | Disables the radio interface if no data activity is detected after the interval defined. Default is 120 seconds. |
|            | inactive-timeout | <minutes>                  | Inactivity interval resulting in the AP terminating its connection. Default is 120 minutes.                      |
|            | snmp             |                            | Sets SNMP access parameters.   |
|            | admin-auth       |                            | Designates a Radius server is used in the authentication verification.   |
|            | server           | <ip>                       | Specifies the IP address the Remote Dial-In User Service (RADIUS) server.  |
|            | port             | <port#>                    | Specifies the port on which the RADIUS server is listening. Default is 1812.                                     |
|            | secret           | <pw>                       | Defines the shared secret password for RADIUS server authentication.   |
|            | mode             | <mode>                     | Enables/disables the access point message mode.  |
|            | msg              |                            | Defines the access point login message text.   |

**admin(system.access)> show**

Displays the current access point access permissions and timeout values.

**Syntax**

**show** Shows all of the current system access settings for the access point..

**Example**

```
admin(system.access)>set trusted-host mode enable
admin(system.access)>set trusted-host range 1 10.1.1.1 10.1.1.10
Warning: Only trusted hosts can access the AP through snmp, http, https, telnet, ssh
```

```
admin(system.access)>show
trusted host access mode          : enable
```

Following trusted host(s) have access to the system via snmp, ssh, http, https and telnet

```
trusted host(s) 1          : 10.1.1.1-10.1.1.10
trusted host(s) 2          : 0.0.0.0-0.0.0.0
trusted host(s) 3          : 0.0.0.0-0.0.0.0
trusted host(s) 4          : 0.0.0.0-0.0.0.0
trusted host(s) 5          : 0.0.0.0-0.0.0.0
trusted host(s) 6          : 0.0.0.0-0.0.0.0
trusted host(s) 7          : 0.0.0.0-0.0.0.0
trusted host(s) 8          : 0.0.0.0-0.0.0.0
```

```
http/s timeout                : 0
ssh server authentication timeout : 120
ssh server inactivity timeout  : 120
admin authentication mode      : local
Login Message Mode             : disable
Login Message                   :
```

**Related Commands:**

**set** Defines the access point system access capabilities and timeout values.

## System Certificate Management Commands

**admin(system)> cmgr**

Navigates to the Certificate Manager submenu. The items available under this command include:

|                    |   |
|--------------------|---|
| <b>genreq</b>      | Generates a Certificate Request.              |
| <b>delfself</b>    | Deletes a Self Certificate.                   |
| <b>loadself</b>    | Loads a Self Certificate signed by CA.        |
| <b>listself</b>    | Lists the self certificate loaded.            |
| <b>loadca</b>      | Loads trusted certificate from CA.            |
| <b>delca</b>       | Deletes the trusted certificate.              |
| <b>listca</b>      | Lists the trusted certificate loaded.         |
| <b>showreq</b>     | Displays a certificate request in PEM format. |
| <b>delpprivkey</b> | Deletes the private key.                      |
| <b>listprivkey</b> | Lists names of private keys.                  |
| <b>expcert</b>     | Exports the certificate file.                 |
| <b>impcert</b>     | Imports the certificate file.                 |
| <b>..</b>          | Goes to the parent menu.                      |
| <b>/</b>           | Goes to the root menu.                        |
| <b>save</b>        | Saves the configuration to system flash.      |
| <b>quit</b>        | Quits the CLI.                                |

**admin(system.cmgr)> genreq**

Generates a certificate request.

**Syntax**

```
genreq <IDname> <Subject> [-ou <OrgUnit>] [-on <OrgName>] [-cn <City>] [-st <State>] . . .
. . . [-p <PostCode>] [-cc <CCode>] [-e <Email>] [-d <Domain>] [-i <IP>] [-sa <SAalgo>]
```

Generates a self-certificate request for a Certification Authority (CA), where:

|               |  |
|---------------|--|
| <IDname>      | The private key ID Name (up to 7 chars)                              |
| <Subject>     | Subject Name (up to 49 chars)  |
| -ou -on       | Organization Unit (up to 49 chars)                                   |
| <OrgName>     | Organization Name (up to 49 chars)                                   |
| -cn <City>    | City Name of Organization (up to 49 chars)                           |
| -st <State>   | State Name (up to 49 chars)  |
| -p <PostCode> | Postal code (9 digits)   |
| -cc <CCode>   | Country code (2 chars)   |
| -e <Email>    | E-mail Address (up to 49 chars)                                      |
| -d <Domain>   | Domain Name (up to 49 chars)   |
| -i <IP>       | IP Address (a.b.c.d)   |
| -sa <SAalgo>  | Signature Algorithm (one of <i>MD5-RSA</i> or <i>SHA1-RSA</i> )      |
| -k <KSize>    | Key size in bits (one of <i>512</i> , <i>1024</i> , or <i>2048</i> ) |

Note: The parameters in [square brackets] are optional. Check with the CA to determine what fields are necessary. For example, most CAs require an email address and an IP address, but not the address of the organization.

**Example**

```
admin(system.cmgr)>genreq MyCert2 MySubject -ou MyDept -on MyCompany
```

```
Please wait. It may take some time...
Generating the certificate request
Retreiving the certificate request
The certificate request is
-----BEGIN CERTIFICATE REQUEST-----
MIHzMIGeAgEAMdkxEjAQBgNVBAoTCU15Q29tcGFueTEPMA0GA1UECzMGTXlEZXB0
MRIwEAYDVQQDEwlNeVN1YmplY3QwXDANBgkqhkiG9w0BAQEFAANLADBIakeEAtKcX
plKFCFAJymTFX71yuxY1fdS7UEhKjBsH7pdqnJnsASK6ZQGAqerjpKScWV1mzYn4
1q2+mgGnCvaZULIo7wIDAQABoAAwDQYJKoZIhvcNAQEEBQADQCClQ5LHdbG/C1f
Bj8AsztSo/bA4dcX3vHvhhJcmuuWO9LHS2imPA3xhX/d6+Q1SMbs+tG4RP0lRSr
iWDyuvwx
-----END CERTIFICATE REQUEST-----
```



## **admin(system.cmgr)> delself**

Deletes a self certificate.

### **Syntax**

**delself**      <IDname>      Deletes the self certificate named <IDname>.

### **Example**

```
admin(system.cmgr)>delself MyCert2
```

**admin(system.cmgr)> loadself**

Loads a self certificate signed by the Certificate Authority.

**Syntax**

**loadself**    <IDname> [https]    Load the self certificate signed by the CA with name <IDname> (7 characters).  
HTTPS is needed for an apacahe certificate and keys.

## **admin(system.cmgr)> listself**

Lists the loaded self certificates.

### **Syntax**

**listself**      Lists all self certificates that are loaded.

**admin(system.cmgr)> loadca**

Loads a trusted certificate from the Certificate Authority.

**Syntax**

**loadca**      Loads the trusted certificate (in PEM format) that is pasted into the command line.

## **admin(system.cmgr)> delca**

Deletes a trusted certificate.

### **Syntax**

**delca**      <IDname>      Deletes the trusted certificate.

**admin(system.cmgr)> listca**

Lists the loaded trusted certificate.

**Syntax**

**listca**            Lists the loaded trusted certificates.

## **admin(system.cmgr)> showreq**

Displays a certificate request in PEM format.

### **Syntax**

**showreq**    <IDname>    Displays a certificate request named <IDname> generated from the genreq command (7 characters maximum).

**admin(system.cmgr)> delprivkey**

Deletes a private key.

**Syntax**

**delprivkey**      <IDname>      Deletes private key named <IDname>.



## **admin(system.cmgr)> listprivkey**

Lists the names of private keys.

### **Syntax**

**listprivkey** Lists all private keys and their associated certificates.

**admin(system.cmgr)> expcert**

Exports the certificate file to a user defined location.

**Syntax**

**expcert**      Exports the access point's CA or Self certificate file.

To export certificate information from an AP3510 or AP3550 model access point:

```
admin(system.cmgr)>expcert ?
```

```
<type> <file name> [https] <cr>      : type: ftp/tftp
                                         : file name: Certificate file name
                                         : https: If set to export apache certificate
                                         : and key
                                         : Server options for this file are the same
                                         : as that for the configuration file
```

```
admin(system.cmgr)>expcert tftp AP-51x1certs.txt
```

To configure AP3510 or AP3550 certificate management settings while conducting a firmware update or restoring a factory default configuration:

```
admin(system.cmgr)> ?
```

```
genreq          : generate a certificate request
delself         : deletes a signed certificate
loadself        : loads a signed certificate signed by the CA
listself        : lists the loaded signed self certificate
loadca          : loads the root CA certificate
delca           : deletes the root CA certificate
listca          : lists the loaded root CA certificate
showreq         : displays certificate request in PEM format
delprivkey      : deletes the private key
listprivkey     : lists the names of the private keys
expcert         : exports the target certificate file
impcert         : imports the target certificate file
(..)            : goes to the parent menu
/               : goes to the root menu
save            : saves the configuration to system flash
quit            : quits the CLI session
```

## **admin(system.cmgr)> impcert**

Imports the target certificate file.

### **Syntax**

**impcert** Imports the target certificate file.

To import certificate information from an AP3510 or AP3550 model access point:

```
admin(system.cmgr)>impcert ?
```

```
<type> <file name> [https] <cr>      : type: ftp/tftp
                                         : file name: Certificate file name
                                         : https: If set to import apache certificate
                                         : and key
                                         : Server options for this file are the same
                                         : as that for the configuration file
```

```
admin(system.cmgr)>impcert tftp AP-51x1certs.txt
```

To configure AP3510 or AP3550 certificate management settings while conducting a firmware update or restoring a factory default configuration:

```
admin(system.cmgr)> ?
```

```
genreq          : generate a certificate request
delsel          : deletes a signed certificate
loadself        : loads a signed certificate signed by the CA
listself        : lists the loaded signed self certificate
loadca          : loads the root CA certificate
delca           : deletes the root CA certificate
listca          : lists the loaded root CA certificate
showreq         : displays certificate request in PEM format
delprivkey      : deletes the private key
listprivkey     : lists the names of the private keys
expcert         : exports the target certificate file
impcert         : imports the target certificate file
(..)           : goes to the parent menu
/              : goes to the root menu
save            : saves the configuration to system flash
quit            : quits the CLI session
```

## System SNMP Commands

### **admin(system)> snmp**

Navigates to the SNMP submenu. The items available under this command are shown below.

|               |  |
|---------------|--|
| <b>access</b> | Goes to the SNMP access submenu.         |
| <b>traps</b>  | Goes to the SNMP traps submenu.          |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |

## System SNMP Access Commands

### **admin(system.snmp)> access**

Navigates to the SNMP Access menu. The items available under this command are shown below.

|               |  |
|---------------|--|
| <b>show</b>   | Shows SNMP v3 engine ID.                 |
| <b>add</b>    | Adds SNMP access entries.                |
| <b>delete</b> | Deletes SNMP access entries.             |
| <b>list</b>   | Lists SNMP access entries.               |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |

**admin(system.snmp.access)> show**

Shows the SNMP v3 engine ID.

**Syntax**

**show**        **eid**        Shows the SNMP v3 Engine ID.

**Example**

```
admin(system.snmp.access)>show eid
```

```
access point snmp v3 engine id                : 000001846B8B4567F871AC68
```

```
admin(system.snmp.access)>
```

## admin(system.snmp.access)> add

Adds SNMP access entries for specific v1v2 and v3 user definitions.

### Syntax

```
add acl      <ip1>    <ip2>    Adds an entry to the SNMP access control list with <ip1> as the starting IP
v1v2c      <comm>    <access>    address and <ip2> and as the ending IP address.
                                         <oid>
                                         : comm - community string 1 to 31 characters
                                         : access - read/write access - (ro,rw)
                                         : oid - string 1 to 127 chars - E.g. 1.3.6.1
v3          <user>    <access>    <oid>          <sec>
          <auth>    <pass1>    <priv>          <pass2>
                                         : user - username 1 to 31 characters
                                         : access - read/write access - (ro,rw)
                                         : oid - string 1 to 127 chars - E.g. 1.3.6.1
                                         : sec - security - (none,auth,auth/priv)
                                         : auth - algorithm - (md5,sha1)
                                         : (required only if sec is - auth,auth/priv)
                                         : pass1 - auth password - 8 to 31 chars
                                         : (required only if sec is 'auth,auth/priv')
                                         : priv - algorithm - (des, aes)
                                         : (required only if sec is 'auth/priv')
                                         : pass2 - privacy password - 8 to 31 chars
                                         : (required only if sec is 'auth/priv')
The following parameters must be specified if <sec> is not none:
    Authentication type <auth> set to md5 or sha1
    Authentication password <pass1> (8 to 31 chars)
The following parameters must be specified if <sec> is set to auth/priv:
    Privacy algorithm set to des or aes
    Privacy password <pass2> (8 to 31 chars)
```

**admin(system.snmp.access)> delete**

Deletes SNMP access entries for specific v1v2 and v3 user definitions.

**Syntax**

|               |       |       |   |
|---------------|-------|-------|---|
| <b>delete</b> | acl   | <idx> | Deletes entry <idx> (1-10) from the access control list.      |
|               |       | all   | Deletes all entries from the access control list.             |
|               | v1v2c | <idx> | Deletes entry <idx> (1-10) from the v1/v2 configuration list. |
|               |       | all   | Deletes all entries from the v1/v2 configuration list.        |
|               | v3    | <idx> | Deletes entry <idx> (1-10) from the v3 user definition list.  |
|               |       | all   | Deletes all entries from the v3 user definition list.         |

**Example**

```
admin(system.snmp.access)>list acl
```

```
-----
index  start ip          end ip
-----
1      209.236.24.1      209.236.24.46
```

```
admin(system.snmp.access)>delete acl all
```

```
admin(system.snmp.access)>list acl
```

```
-----
index  start ip          end ip
-----
```



## admin(system.snmp.access)> list

Lists SNMP access entries.

### Syntax

|             |       |   |
|-------------|-------|---|
| <b>list</b> | acl   | Lists SNMP access control list entries.               |
|             | v1v2c | Lists SNMP v1/v2c configuration.                      |
|             | v3    | <idx> Lists SNMP v3 user definition with index <idx>. |
|             | all   | Lists all SNMP v3 user definitions.                   |

### Example

```
admin(system.snmp.access)>list acl
```

```
-----  
index  start ip      end ip  
-----  
1      209.236.24.1    209.236.24.46
```

```
admin(system.snmp.access)>list v1v2c
```

```
-----  
index  community      access      oid  
-----  
1      public          read only   1.3.6.1  
2      private         read/write  1.3.6.1
```

```
admin(system.snmp.access)>list v3 2
```

```
index           : 2  
username         : judy  
access permission : read/write  
object identifier : 1.3.6.1  
security level   : auth/priv  
auth algorithm   : md5  
auth password    : *****  
privacy algorithm : des  
privacy password : *****
```

## System SNMP Traps Commands

### **admin(system.snmp)> traps**

Navigates to the SNMP traps submenu. The items available under this command are shown below.

|               |  |
|---------------|--|
| <b>show</b>   | Shows SNMP trap parameters.              |
| <b>set</b>    | Sets SNMP trap parameters.               |
| <b>add</b>    | Adds SNMP trap entries.                  |
| <b>delete</b> | Deletes SNMP trap entries.               |
| <b>list</b>   | Lists SNMP trap entries.                 |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |

## **admin(system.snmp.traps)> show**

Shows SNMP trap parameters.

### **Syntax**

|             |           |  |
|-------------|-----------|--|
| <b>show</b> | trap      | Shows SNMP trap parameter settings.      |
|             | rate-trap | Shows SNMP rate-trap parameter settings. |

### **Example**

```
admin(system.snmp.traps)>show trap
```

```
SNMP MU Traps
  mu associated           : enable
  mu unassociated         : disable
  mu denied association   : disable
  mu denied authentication : disable

SNMP Traps
  snmp authentication failure : disable
  snmp acl violation          : disable

SNMP Network Traps
  physical port status change : enable
  denial of service           : enable
  denial of service trap rate limit : 10 seconds

SNMP System Traps
  system cold start         : disable
  system config changed     : disable
  rogue ap detection        : disable
  ap radar detection        : disable
  wpa counter measure       : disable
  mu hotspot status         : disable
  vlan                     : disable
  lan monitor               : disable
  DynDNS Update             : enable
```

**admin(system.snmp.traps)> set**

Sets SNMP trap parameters.

**Syntax**

|            |                   |                |         |         |  |
|------------|-------------------|----------------|---------|---------|--|
| <b>set</b> | mu-assoc          | enable/disable |         |         | Enables/disables the Client associated trap.   |
|            | mu-unassoc        | enable/disable |         |         | Enables/disables the Client unassociated trap.   |
|            | mu-deny-assoc     | enable/disable |         |         | Enables/disables the Client association denied trap.   |
|            | mu-deny-auth      | enable/disable |         |         | Enables/disables the Client authentication denied trap.  |
|            | snmp-auth         | enable/disable |         |         | Enables/disables the authentication failure trap.  |
|            | snmp-acl          | enable/disable |         |         | Enables/disables the SNMP ACL violation trap.  |
|            | port              | enable/disable |         |         | Enables/disables the physical port status trap.  |
|            | dos-attack        | enable/disable |         |         | Enables/disables the denial of service trap.   |
|            | dyndns-update     | enable/disable |         |         | Enables/disables dyndns update trap.   |
|            | interval          | <rate>         |         |         | Sets denial of service trap interval.  |
|            | cold              | enable/disable |         |         | Enables/disables the system cold start trap.   |
|            | cfg               | enable/disable |         |         | Enables/disables a configuration changes trap.   |
|            | rogue-ap          | enable/disable |         |         | Enables/disables a trap when a rogue-ap is detected.   |
|            | ap-radar          | enable/disable |         |         | Enables/disables the AP Radar Detection trap.  |
|            | wpa-counter       | enable/disable |         |         | Enables/disables the WPA counter measure trap.   |
|            | hotspot-mu-status | enable/disable |         |         | Enables/disables the hotspot Client status trap.   |
|            | vlan              | enable/disable |         |         | Enables/disables VLAN traps.   |
|            | lan-monitor       | enable/disable |         |         | Enables/disables LAN monitor traps.  |
|            | rate              | <rate>         | <scope> | <value> | Sets the particular <rate> to monitor to <value> given the indicated <scope>. See table below for information on the possible values for <rate>, <scope>, and <value>. |
|            | min-pkt           | <pkt>          |         |         | Sets the minimum number of packets required for rate traps to fire (1-65535).  |

## admin(system.snmp.traps)> add

Adds SNMP trap entries.

### Syntax

**add** v1v2 <ip> <port> <comm> <ver>  
Adds an entry to the SNMP v1/v2 access list with the destination IP address set to <ip>, the destination UDP port set to <port>, the community string set to <comm> (1 to 31 characters), and the SNMP version set to <ver>.

v3 <ip> <port> <user> <sec> <auth> <pass1> <priv> <pass2>  
Adds an entry to the SNMP v3 access list with the destination IP address set to <ip>, the destination UDP port set to <port>, the username set to <user> (1 to 31 characters), and the authentication type set to one of *none*, *auth*, or *auth/priv*.

The following parameters must be specified if <sec> is not *none*:  
Authentication type <auth> set to *md5* or *sha1*  
Authentication password <pass1> (8 to 31 chars)

The following parameters must be specified if <sec> is set to *auth/priv*:  
Privacy algorithm set to *des* or *aes*  
Privacy password <pass2> (8 to 31 chars)

### Example

```
admin(system.snmp.traps)>add v1v2 203.223.24.2 333 mycomm v1
admin(system.snmp.traps)>list v1v2c
```

| index | dest ip      | dest port | community | version |
|-------|--------------|-----------|-----------|---------|
| 1     | 203.223.24.2 | 333       | mycomm    | v1      |

```
admin(system.snmp.traps)>add v3 201.232.24.33 555 BigBoss none md5
admin(system.snmp.traps)>list v3 all
```

|                   |                 |
|-------------------|-----------------|
| index             | : 1             |
| destination ip    | : 201.232.24.33 |
| destination port  | : 555           |
| username          | : BigBoss       |
| security level    | : none          |
| auth algorithm    | : md5           |
| auth password     | : *****         |
| privacy algorithm | : des           |
| privacy password  | : *****         |

**admin(system.snmp.traps)> delete**

Deletes SNMP trap entries.

**Syntax**

|               |       |       |   |
|---------------|-------|-------|---|
| <b>delete</b> | v1v2c | <idx> | Deletes entry <idx> from the v1v2c access control list. |
|               |       | all   | Deletes all entries from the v1v2c access control list. |
|               | v3    | <idx> | Deletes entry <idx> from the v3 access control list.    |
|               |       | all   | Deletes all entries from the v3 access control list.    |

**Example**

```
admin(system.snmp.traps)>delete v1v2 all
```

## admin(system.snmp.traps)> list

Lists SNMP trap entries.

### Syntax

**list**    v1v2c               Lists SNMP v1/v2c access entries.  
         v3                <idx>       Lists SNMP v3 access entry <idx>.  
         all               Lists all SNMP v3 access entries.

### Example

```
admin(system.snmp.traps)>add v1v2 203.223.24.2 162 mycomm v1
admin(system.snmp.traps)>list v1v2c
```

| index | dest ip      | dest port | community | version |
|-------|--------------|-----------|-----------|---------|
| 1     | 203.223.24.2 | 162       | mycomm    | v1      |

```
admin(system.snmp.traps)>add v3 201.232.24.33 555 BigBoss none md5
admin(system.snmp.traps)>list v3 all
```

|                   |                 |
|-------------------|-----------------|
| index             | : 1             |
| destination ip    | : 201.232.24.33 |
| destination port  | : 555           |
| username          | : BigBoss       |
| security level    | : none          |
| auth algorithm    | : md5           |
| auth password     | : *****         |
| privacy algorithm | : des           |
| privacy password  | : *****         |

## System User Database Commands

### **admin(system)> userdb**

Navigates to the user database submenu.

### **Syntax**

|              |  |
|--------------|--|
| <b>user</b>  | Goes to the user submenu.                |
| <b>group</b> | Goes to the group submenu.               |
| <b>save</b>  | Saves the configuration to system flash. |
| <b>..</b>    | Goes to the parent menu.                 |
| <b>/</b>     | Goes to the root menu.                   |



## Adding and Removing Users from the User Database

### **admin(system.userdb)> user**

Adds and removes users from the user database and defines user passwords.

### **Syntax**

|                 |   |
|-----------------|---|
| <b>add</b>      | Adds a new user.                                  |
| <b>delete</b>   | Deletes an existing user ID..                     |
| <b>clearall</b> | Removes all existing user IDs from the system.    |
| <b>set</b>      | Sets a password for a user.                       |
| <b>show</b>     | Displays the current user database configuration. |
| <b>save</b>     | Saves the configuration to system flash.          |
| <b>..</b>       | Goes to the parent menu.                          |
| <b>/</b>        | Goes to the root menu.                            |

**admin(system.userdb.user)> add**

Adds a new user to the user database.

**Syntax**

**add**            <name>        Adds a new user and password to the user database.  
                 <password>

**Example**

```
admin(system.userdb.user>add george password
```

```
admin(system.userdb.user>
```

## **admin(system.userdb.user)> delete**

Removes a new user to the user database.

### **Syntax**

**delete**                Removes a user ID string from the user database.

### **Example**

```
admin(system.userdb.user>delete george
```

```
admin(system.userdb.user>
```

**admin(system.userdb.user)> clearall**

Removes all existing user IDs from the system.

**Syntax**

**clearall**                Removes all existing user IDs from the system.

**Example**

```
admin(system.userdb.user>clearall
```

```
admin(system.userdb.user>
```

## **admin(system.userdb.user)> set**

Sets a password for a user..

### **Syntax**

|            |          |                                      |
|------------|----------|--------------------------------------|
| <b>set</b> | <userid> | Sets a password for a specific user. |
|            | <passwd> |                                      |

### **Example**

```
admin(system.userdb.user>set george password
```

```
admin(system.userdb.user>
```

## Adding and Removing Groups from the User Database

### **admin(system.userdb)> group**

Adds or removes groups from the user database.

#### **Syntax**

|                 |   |
|-----------------|---|
| <b>create</b>   | Creates a group name.                             |
| <b>delete</b>   | Deletes a group name.                             |
| <b>clearall</b> | Removes all existing group names from the system. |
| <b>add</b>      | Adds a user to an existing group.                 |
| <b>remove</b>   | Removes a user from an existing group.            |
| <b>show</b>     | Displays existing groups.                         |
| <b>save</b>     | Saves the configuration to system flash.          |
| <b>..</b>       | Goes to the parent menu.                          |
| <b>/</b>        | Moves back to root menu.                          |

## **admin(system.userdb.group)> create**

Creates a group name. Once defined, users can be added to the group.

### **Syntax**

**create**                      Creates a group name. Once defined, users can be added to the group.

### **Example**

```
admin(system.userdb.group>create 2
```

```
admin(system.userdb.group>
```

**admin(system.userdb.group)> delete**

Deletes an existing group.

**Syntax**

**delete**                Deletes an existing group.

**Example**

```
admin(system.userdb.group>delete 2
```

```
admin(system.userdb.group>
```



## **admin(system.userdb.group)> clearall**

Removes all existing group names from the system.

### **Syntax**

**clearall**                Removes all existing group names from the system.

### **Example**

```
admin(system.userdb.group>clearall
```

```
admin(system.userdb.group>
```

**admin(system.userdb.group)> add**

Adds a user to an existing group.

**Syntax**

**add** <userid> <group> Adds a user <userid> to an existing group <group>.

**Example**

```
admin(system.userdb.group>add lucy group x
```

```
admin(system.userdb.group>
```

## **admin(system.userdb.group)> remove**

Removes a user from an existing group.

### **Syntax**

**remove**                    <userid> <group> Removes a user <userid> from an existing group<group> .

### **Example**

```
admin(system.userdb.group>remove lucy group x
```

```
admin(system.userdb.group>
```

**admin(system.userdb.group)> show**

Displays existing groups.

**Syntax**

|             |        |   |
|-------------|--------|---|
| <b>show</b> |        | Displays existing groups and users.       |
|             | users  | Displays configured user IDs for a group. |
|             | groups | Displays configured groups.               |

**Example**

```
admin(system.userdb.group>show groups
```

List of Group Names

```
      : engineering
      : marketing
      : demo room
```

```
admin(system.userdb.group>
```

## System Radius Commands

### **admin(system)> radius**

Navigates to the Radius system submenu.

### **Syntax**

|               |  |
|---------------|--|
| <b>eap</b>    | Goes to the EAP submenu.                 |
| <b>policy</b> | Goes to the access policy submenu.       |
| <b>ldap</b>   | Goes to the LDAP submenu.                |
| <b>proxy</b>  | Goes to the proxy submenu.               |
| <b>client</b> | Goes to the client submenu.              |
| <b>set</b>    | Sets Radius parameters.                  |
| <b>show</b>   | Displays Radius parameters.              |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |

**admin(system.radius)> set/show**

Sets or displays the Radius user database.

**Syntax**

**set**                      Sets the Radius user database.  
**show all**                Displays the Radius user database.

**Example**

```
admin(system.radius)>set database local  
admin(system.radius)>show all
```

```
Database                      : local
```

```
admin(system.radius)>
```

## **admin(system.radius)> eap**

Navigates to the EAP submenu.

### **Syntax**

|               |  |
|---------------|--|
| <b>peap</b>   | Goes to the Peap submenu.                |
| <b>ttls</b>   | Goes to the TTLS submenu.                |
| <b>import</b> | Imports the requested EAP certificates.  |
| <b>set</b>    | Defines EAP parameters.                  |
| <b>show</b>   | Displays the EAP configuration.          |
| <b>save</b>   | Saves the configuration to system flash. |
| <b>quit</b>   | Quits the CLI.                           |
| <b>..</b>     | Goes to the parent menu.                 |
| <b>/</b>      | Goes to the root menu.                   |

**admin(system.radius.eap)> peap**

Navigates to the Peap submenu.

**Syntax**

|             |  |
|-------------|--|
| <b>set</b>  | Defines Peap parameters.                 |
| <b>show</b> | Displays the Peap configuration.         |
| <b>save</b> | Saves the configuration to system flash. |
| <b>quit</b> | Quits the CLI.                           |
| <b>..</b>   | Goes to the parent menu.                 |
| <b>/</b>    | Goes to the root menu.                   |



```
admin(system.radius.eap.peap)> set/show
```

Defines and displays Peap parameters

## Syntax

|             |  |
|-------------|--|
| <b>set</b>  | Sets the Peap authentication <type>.   |
| <b>show</b> | Displays the Peap authentication type. |

### Example

```
admin(system.radius.eap.peap)>set auth gtc
admin(system.radius.eap.peap)>show
```

```
PEAP Auth Type      : gtc
```

**admin(system.radius.eap)> ttls**

Navigates to the TTLS submenu.

**Syntax**

|             |  |
|-------------|--|
| <b>set</b>  | Defines TTLS parameters.                 |
| <b>show</b> | Displays the TTLS configuration.         |
| <b>save</b> | Saves the configuration to system flash. |
| <b>quit</b> | Quits the CLI.                           |
| <b>..</b>   | Goes to the parent menu.                 |
| <b>/</b>    | Goes to the root menu.                   |

```
admin(system.radius.eap.ttls)> set/show
```

Defines and displays TLS parameters

## Syntax

|             |  |
|-------------|--|
| <b>set</b>  | Sets the TTLS authentication <type>.   |
| <b>show</b> | Displays the TTLS authentication type. |

### Example

```
admin(system.radius.eap.ttls)>set auth pap
admin(system.radius.eap.ttls)>show
```

TTL Auth Type : pap

**admin(system.radius)> policy**

Navigates to the access policy submenu.

**Syntax**

|                    |  |
|--------------------|--|
| <b>set</b>         | Sets a group's WLAN access policy.       |
| <b>access-time</b> | Goes to the time based login submenu.    |
| <b>show</b>        | Displays the group's access policy.      |
| <b>save</b>        | Saves the configuration to system flash. |
| <b>quit</b>        | Quits the CLI.                           |
| <b>..</b>          | Goes to the parent menu.                 |
| <b>/</b>           | Goes to the root menu.                   |

## **admin(system.radius.policy)> set**

Defines the group's WLAN access policy.

### **Syntax**

|            |                              |  |
|------------|------------------------------|--|
| <b>set</b> | <code>&lt;group&gt;</code>   | Defines the group's <group name> WLAN access policy (WLAN name delimited |
|            | <code>&lt;wlan(s)&gt;</code> | by a space).   |

### **Example**

```
admin(system.radius.policy)>set engineering 16
```

```
admin(system.radius.policy)>
```

**admin(system.radius.policy)> access-time**

Goes to the time-based login submenu.

**Syntax**

```

set           <group>      Defines a target group's access time permissions. Access time is in DayDDDD-
                  <access-time> DDDD format.
show          Displays the group's access time rule.
save          Saves the configuration to system flash.
quit          Quits the CLI.
..           Goes to the parent menu.
/            Goes to the root menu.

```

**Example**

```
admin(system.radius.policy.access-time)>show
```

List of Access Policies

```

1           : Tue0830-2200, We2000-2300, Th1100-1930
2           : Any0000-2359
10          : Any0000-2359
12          : Any0000-2359

```

| Context                          | Command   | Description   |
|----------------------------------|---|---|
| system>radius>policy>access-time | set start-time <group><br><value>                 | group = Valid group name.<br>value = 4 digit value<br>representing HHMM<br>(0000-2359 allowed).   |
| system>radius>policy>access-time | set end-time <group><br><value>                   | group = Valid group name.<br>value = 4 digit value<br>representing HHMM<br>(0000-2359 allowed).<br><br>The end time should be<br>greater than the start time. |
| system>radius>policy>access-time | set access-days <group><br><day-selector-keyword> | group = Valid group name.<br>day-selector-keyword = The<br>allowed values are:<br>Mo, Tu, We, Th, Fr, Sa, Su,<br>Weekdays, Weekends, all.                     |

## **admin(system.radius.policy)> show**

Displays a group's access policy.

### **Syntax**

**show**                      Displays a group's access policy.

### **Example**

```
admin(system.radius.policy)>show
```

List of Access Policies

|             |            |
|-------------|------------|
| engineering | : 16       |
| marketing   | : 10       |
| demo room   | : 3        |
| test demo   | : No Wlans |

```
admin(system.radius.policy)>
```

**admin(system.radius)> ldap**

Navigates to the LDAP submenu.

**Syntax**

|             |   |
|-------------|---|
| <b>set</b>  | Defines the LDAP parameters.  |
| <b>show</b> | Displays existing LDAP parameters (command must be supplied as “show all.”) |
| <b>save</b> | Saves the configuration to system flash.                                    |
| <b>quit</b> | Quits the CLI.  |
| <b>..</b>   | Goes to the parent menu.  |
| <b>/</b>    | Goes to the root menu.  |



## **admin(system.radius.ldap)> set**

Defines the LDAP parameters.

### **Syntax**

|            |                                       |
|------------|---------------------------------------|
| <b>set</b> | Defines the LDAP parameters.          |
| ipadr      | Sets LDAP IP address.                 |
| port       | Sets LDAP server port.                |
| binddn     | Sets LDAP bind distinguished name.    |
| basedn     | Sets LDAP base distinguished name.    |
| passwd     | Sets LDAP server password.            |
| login      | Sets LDAP login attribute.            |
| pass_attr  | Sets LDAP password attribute.         |
| groupname  | Sets LDAP group name attribute.       |
| filter     | Sets LDAP group membership filter.    |
| membership | Sets LDAP group membership attribute. |

### **Example**

```
admin(system.radius.ldap)>set ipadr 157.235.121.12
admin(system.radius.ldap)>set port 203.21.37.18
admin(system.radius.ldap)>set binddn 123
admin(system.radius.ldap)>set basedn 203.21.37.19
admin(system.radius.ldap)>set passwd mudskipper
admin(system.radius.ldap)>set login muddy
admin(system.radius.ldap)>set pass_attr 123
admin(system.radius.ldap)>set groupname 0.0.0.0
admin(system.radius.ldap)>set filter 123
admin(system.radius.ldap)>set membership radiusGroupName

admin(system.radius.ldap)>
```

**admin(system.radius.ldap)> show all**

Displays existing LDAP parameters.

**Syntax**

**show all**      Displays existing LDAP parameters.

**Example**

```
admin(system.radius.ldap)>show all
```

```
LDAP Server IP           : 0.0.0.0
LDAP Server Port         : 389
LDAP Bind DN             : cn=manager, o=trion
LDAP Base DN             : 0=trion
LDAP Login Attribute     : (uid=%{Stripped-User-Name:-%{User-Name}})
LDAP Password attribute  : userPassword
LDAP Group Name Attribute : cn
LDAP Group Membership Filter : (|(&(objectClass=GroupOfNames)(member=%{Ldap-
objectClass=GroupOfUniqueNames)(uniquemember=%{Ldap-UserDn})))
LDAP Group Membership Attribute : radiusGroupName
```

```
admin(system.radius.ldap)>
```

## **admin(system.radius)> proxy**

Navigates to the Radius proxy server submenu.

### **Syntax**

|                 |  |
|-----------------|--|
| <b>add</b>      | Adds a proxy realm.                              |
| <b>delete</b>   | Deletes a proxy realm.                           |
| <b>clearall</b> | Removes all proxy server records.                |
| <b>set</b>      | Sets proxy server parameters.                    |
| <b>show</b>     | Displays current Radius proxy server parameters. |
| <b>save</b>     | Saves the configuration to system flash.         |
| <b>quit</b>     | Quits the CLI.                                   |
| <b>..</b>       | Goes to the parent menu.                         |
| <b>/</b>        | Goes to the root menu.                           |

**admin(system.radius.proxy)> add**

Adds a proxy.

**Syntax**

|            |      |        |                                   |
|------------|------|--------|-----------------------------------|
| <b>add</b> |      |        | Adds a proxy realm.               |
|            | name | <name> | Realm name.                       |
|            | ip1  | <ip1>  | Authentication server IP address. |
|            | port | <port> | Authentication server port.       |
|            | sec  | <sec>  | Shared secret password.           |

**Example**

```
admin(system.radius.proxy)>add lance1ot 157.235.241.22 1812 muddy
```

```
admin(system.radius.proxy)>
```

## **admin(system.radius.proxy)> delete**

Adds a proxy.

### **Syntax**

**delete**            <realm>       Deletes a specified realm name.

### **Example**

```
admin(system.radius.proxy)>delete lancelot
```

```
admin(system.radius.proxy)>
```

**admin(system.radius.proxy)> clearall**

Removes all proxy server records from the system.

**Syntax**

**clearall**      Removes all proxy server records from the system.

**Example**

```
admin(system.radius.proxy)>clearall
```

```
admin(system.radius.proxy)>
```

## **admin(system.radius.proxy)> set**

Sets Radius proxy server parameters.

### **Syntax**

|            |       |   |
|------------|-------|---|
| <b>set</b> |       | Sets Radius proxy server parameters.                        |
|            | delay | Defines retry delay time (in seconds) for the proxy server. |
|            | count | Defines retry count value for the proxy server.             |

### **Example**

```
admin(system.radius.proxy)>set delay 10
```

```
admin(system.radius.proxy)>set count 5
```

```
admin(system.radius.proxy)>
```

**admin(system.radius)> client**

Goes to the Radius client submenu.

**Syntax**

|               |   |
|---------------|---|
| <b>add</b>    | Adds a Radius client to list of available clients.      |
| <b>delete</b> | Deletes a Radius client from list of available clients. |
| <b>show</b>   | Displays a list of configured clients.                  |
| <b>save</b>   | Saves the configuration to system flash.                |
| <b>quit</b>   | Quits the CLI.  |
| <b>..</b>     | Goes to the parent menu.                                |
| <b>/</b>      | Goes to the root menu.                                  |



## **admin(system.radius.client)> add**

Adds a Radius client to those available to the Radius server.

### **Syntax**

|            |        |       |                                     |
|------------|--------|-------|-------------------------------------|
| <b>add</b> |        |       | Adds a proxy.                       |
|            | ip     | <ip>  | Client's IP address.                |
|            | mask   | <ip1> | Network mask address of the client. |
|            | secret | <sec> | Shared secret password.             |

### **Example**

```
admin(system.radius.client)>add 157.235.132.11 255.255.255.225 muddy
```

```
admin(system.radius.client)>
```

**admin(system.radius.client)> delete**

Removes a specified Radius client from those available to the Radius server.

**Syntax**

|               |         |   |
|---------------|---------|---|
| <b>delete</b> | <ipadr> | Removes a specified Radius client (by IP address) from those available to the Radius server |
|---------------|---------|---|

**Example**

```
admin(system.radius.client)>delete 157.235.132.11
```

```
admin(system.radius.client)>
```

## **admin(system.radius.client)> show**

Displays a list of configured Radius clients.

### **Syntax**

**show**                Removes a specified Radius client from those available to the Radius server.

### **Example**

```
admin(system.radius.client)>show
```

| Idx | Subnet/Host    | Netmask         | SharedSecret |
|-----|----------------|-----------------|--------------|
| 1   | 157.235.132.11 | 255.255.255.225 | *****        |

```
admin(system.radius.client)>
```

## System Network Time Protocol (NTP) Commands

### **admin(system)> ntp**

Navigates to the NTP menu. The correct network time is required for numerous functions to be configured accurately on the access point.

### **Syntax**

|                  |  |
|------------------|--|
| <b>show</b>      | Shows NTP parameters settings.           |
| <b>date-zone</b> | Show date, time and time zone.           |
| <b>zone-list</b> | Displays list of time zones.             |
| <b>set</b>       | Sets NTP parameters.                     |
| <b>..</b>        | Goes to the parent menu.                 |
| <b>/</b>         | Goes to the root menu.                   |
| <b>save</b>      | Saves the configuration to system flash. |
| <b>quit</b>      | Quits the CLI.                           |

## **admin(system.ntp)> show**

Displays the NTP server configuration.

### **Syntax**

**show**      Shows all NTP server settings.

### **Example**

```
admin(system.ntp)>show
```

```
current time (UTC)                : 2006-07-31 14:35:20
```

Time Zone:

```
ntp mode                          : enable
preferred Time server ip          : 203.21.37.18
preferred Time server port        : 123
first alternate server ip         : 203.21.37.19
first alternate server port        : 123
second alternate server ip        : 0.0.0.0
second alternate server port      : 123
synchronization interval         : 15 minutes
```

**admin(system.ntp)> date-zone**

Show date, time and time zone.

**Syntax**

**date-zone**            Show date, time and time zone.

**Example**

```
admin(system.ntp)> date-zone
```

```
Date/Time                : Sat 1970-Jan-03 20:06:22 +0000 UTC
```

```
Time Zone                : UTC
```

## **admin(system.ntp)> zone-list**

Displays an extensive list of time zones for countries around the world.

### **Syntax**

**zone-list**                Displays list of time zone indexes for every known zone.

### **Example**

```
admin(system.ntp)>date-zone
```

**admin(system.ntp)> set**

Sets NTP parameters for access point clock synchronization.

**Syntax**

|            |        |              |  |
|------------|--------|--------------|--|
| <b>set</b> | mode   | <ntp-mode>   | Enables or disables NTP.   |
|            | server | <idx> <ip>   | Sets the NTP sever IP address.   |
|            | port   | <idx> <port> | Defines the port number.   |
|            | intrvl | <period>     | Defines the clock synchronization interval used between the access point and the NTP server in minutes (15 - 65535).   |
|            | time   | <time>       | Sets the current system time. [yyyy] - year, [mm] - month, [dd] - day of the month, [hh] - hour of the day, [mm] - minute, [ss] second, [zone -idx] Index of the zone. |
|            | zone   | <zone>       | Defines the time zone (by index) for the target country.   |

**Example**

```
admin(system.ntp)>set mode enable
admin(system.ntp)>set server 1 203.21.37.18
admin(system.ntp)>set port 1 123
admin(system.ntp)>set intrvl 15
admin(system.ntp)>set zone 1
```



## System Log Commands

### **admin(system)> logs**

Navigates to the access point log submenu. Logging options include:

#### **Syntax**

|               |   |
|---------------|---|
| <b>show</b>   | Shows logging options.                  |
| <b>set</b>    | Sets log options and parameters.        |
| <b>view</b>   | Views system log.                       |
| <b>delete</b> | Deletes the system log.                 |
| <b>send</b>   | Sends log to the designated FTP Server. |
| <b>..</b>     | Goes to the parent menu.                |
| <b>/</b>      | Goes to the root menu.                  |
| <b>save</b>   | Saves configuration to system flash.    |
| <b>quit</b>   | Quits the CLI.                          |

**admin(system.logs)> show**

Displays the current access point logging settings.

**Syntax**

**show**                Displays the current access point logging configuration.

**Example**

```
admin(system.logs)>show
```

```
log level                : L6 Info
syslog server logging    : enable
syslog server ip address : 192.168.0.102
```

## **admin(system.logs)> set**

Sets log options and parameters.

### **Syntax**

|            |       |           |   |
|------------|-------|-----------|---|
| <b>set</b> | level | <level>   | Sets the level of the events that will be logged. All events with a level at or above <level> (L0-L7) will be saved to the system log.<br>L0:Emergency<br>L1:Alert<br>L2:Critical<br>L3:Errors<br>L4:Warning<br>L5:Notice<br>L6:Info ( <i>default setting</i> )<br>L7:Debug |
|            | mode  | <op-mode> | Enables or disables syslog server logging.  |
|            | ipadr | <ip>      | Sets the external syslog server IP address to <ip> (a.b.c.d).   |

```
admin(system.logs)>set mode enable
```

```
admin(system.logs)>set level L4
```

```
admin(system.logs)>set ipadr 157.235.112.11
```

**admin(system.logs)> view**

Displays the access point system log file.

**Syntax**

**view**            Displays the entire access point system log file.

**Example**

```
admin(system.logs)>view
```

```
Jan  7 16:14:00 (none) syslogd 1.4.1: restart (remote reception).
Jan  7 16:14:10 (none) klogd: :ps log:fc: queue maintenance
Jan  7 16:14:41 (none) klogd: :ps log:fc: queue maintenance
Jan  7 16:15:43 (none) last message repeated 2 times
Jan  7 16:16:01 (none) CC:   4:16pm  up 6 days, 16:16, load average: 0.00, 0.01,
0.00
Jan  7 16:16:01 (none) CC:   Mem:           62384           32520           29864
0
0
Jan  7 16:16:01 (none) CC: 0000077e 0012e95b 0000d843 00000000 00000003 0000121
e 00000000 00000000 0037ebf7 000034dc 00000000 00000000 00000000
Jan  7 16:16:13 (none) klogd: :ps log:fc: queue maintenance
Jan  7 16:16:44 (none) klogd: :ps log:fc: queue maintenance
Jan  7 16:17:15 (none) klogd: :ps log:fc: queue maintenance
Jan  7 16:17:15 (none) klogd: :ps log:fc: queue maintenance
```

## **admin(system.logs)> delete**

Deletes the log files.

### **Syntax**

**delete**     Deletes the access point system log file.

### **Example**

```
admin(system.logs)>delete
```

**admin(system.logs)> send**

Sends log and core file to an FTP Server.

**Syntax**

**send**        Sends the system log file via FTP to a location specified with the set command. Refer to the command set under the (system.fwupdate) command for information on setting up an FTP server and login information.

**Example**

```
admin(system.logs)>send
```

```
File transfer           : [ In progress ]
File transfer           : [ Done ]
```

```
admin(system.logs)>
```

## System Configuration-Update Commands

### **admin(system)> config**

Navigates to the access point configuration update submenu.

### **Syntax**

|                |  |
|----------------|--|
| <b>default</b> | Restores the default access point configuration.           |
| <b>partial</b> | Restores a partial default access point configuration.     |
| <b>show</b>    | Shows import/export parameters.                            |
| <b>set</b>     | Sets import/export access point configuration parameters.  |
| <b>export</b>  | Exports access point configuration to a designated system. |
| <b>import</b>  | Imports configuration to the access point.                 |
| <b>..</b>      | Goes to the parent menu.                                   |
| <b>/</b>       | Goes to the root menu.                                     |
| <b>save</b>    | Saves the configuration to access point system flash.      |
| <b>quit</b>    | Quits the CLI.   |

**admin(system.config)> default**

Restores the full access point factory default configuration.

**Syntax**

**default**       Restores the access point to the original (factory) configuration.

**Example**

```
admin(system.config)>default
```

```
Are you sure you want to default the configuration? <yes/no>:
```



## **admin(system.config)> partial**

Restores a partial factory default configuration. The access point's LAN, WAN and SNMP settings are unaffected by the partial restore.

### **Syntax**

**default**       Restores a partial access point configuration.

### **Example**

```
admin(system.config)>partial
```

```
Are you sure you want to partially default AP35xx? <yes/no>:
```

**admin(system.config)> show**

Displays import/export parameters for the access point configuration file.

**Syntax**

**show**       Shows all import/export parameters.

**Example**

```
admin(system.config)>show
```

```
cfg filename           : cfg.txt
cfg filepath           :
ftp/tftp server ip address : 192.168.0.101
ftp user name          : myadmin
ftp password           : *****
```

## **admin(system.config)> set**

Sets the import/export parameters.

### **Syntax**

|            |        |             |  |
|------------|--------|-------------|--|
| <b>set</b> | file   | <filename>  | Sets the configuration file name (1 to 39 characters in length). |
|            | path   | <path>      | Defines the path used for the configuration file upload.         |
|            | server | <ipaddress> | Sets the FTP/TFTP server IP address.                             |
|            | user   | <username>  | Sets the FTP user name (1 to 39 characters in length).           |
|            | passwd | <pswd>      | Sets the FTP password (1 to 39 characters in length).            |

### **Example**

```
admin(system.config)>set server 192.168.22.12
```

```
admin(system.config)>set user myadmin
```

```
admin(system.config)>set passwd georges
```

```
admin(system.config)>show
```

```
cfg filename           : cfg.txt
cfg filepath           :
ftp/tftp server ip address : 192.168.22.12
ftp user name          : myadmin
ftp password           : *****
```

**admin(system.config)> export**

Exports the configuration from the system.

**Syntax**

|               |          |   |
|---------------|----------|---|
| <b>export</b> | ftp      | Exports the access point configuration to the FTP server. Use the set command to set the server, user, password, and file name before using this command. |
|               | tftp     | Exports the access point configuration to the TFTP server. Use the set command to set the IP address for the TFTP server before using the command.        |
|               | terminal | Exports the access point configuration to a terminal.   |

**Example**

Export FTP

```
admin(system.config)>set server 192.168.22.12
admin(system.config)>set user myadmin
admin(system.config)>set file config.txt
admin(system.config)>set passwd
```

```
admin(system.config)>export ftp
```

```
Export operation           : [ Started ]
Building configuration file : [ Done ]
File transfer              : [ In progress ]
File transfer              : [ Done ]
Export Operation           : [ Done ]
```

**Example**

Export TFTP

```
admin(system.config)>set server 192.168.0.101
admin(system.config)>set file config.txt
admin(system.config)>export tftp
```

```
Export operation           : [ Started ]
Building configuration file : [ Done ]
File transfer              : [ In progress ]
File transfer              : [ Done ]
Export Operation           : [ Done ]
```

**CAUTION**

***Make sure a copy of the access point's current configuration is exported (to a secure location) before exporting the access point's configuration, as you will want a valid version available in case errors are encountered with the configuration export.***

## admin(system.config)> import

Imports the access point configuration to the access point. Errors could display as a result of invalid configuration parameters. Correct the specified lines and import the file again until the import operation is error free.

### Syntax

|               |      |   |
|---------------|------|---|
| <b>import</b> | ftp  | Imports the access point configuration file from the FTP server. Use the set command to set the server, user, password, and file. |
|               | tftp | Imports the access point configuration from the TFTP server. Use the set command to set the server and file.                      |

### Example

#### Import FTP Example

```
admin(system.config)>set server 192.168.22.12
admin(system.config)>set user myadmin
admin(system.config)>set file config.txt
admin(system.config)>set passwd mysecret
admin(system.config)>import ftp
Import operation : [ Started ]
File transfer : [ In progress ]
File transfer : [ Done ]
Import operation : [ Done ]
```

#### Import TFTP Example

```
admin(system.config)>set server 192.168.0.101
admin(system.config)>set file config.txt
admin(system.config)>import tftp
Import operation : [ Started ]
File transfer : [ In progress ]
File transfer : [ Done ]
Import operation : [ Done ]
```



### CAUTION

***A single-radio model access point cannot import/export its configuration to a dual-radio model access point. In turn, a dual-radio model access point cannot import/export its configuration to a single-radio access point.***

## Firmware Update Commands

### **admin(system)> fw-update**

Navigates to the firmware update submenu. The items available under this command are shown below.



#### **NOTE**

*The access point must complete the reboot process to successfully update the device firmware, regardless of whether the reboot is conducted using the GUI or CLI interfaces.*

|               |   |
|---------------|---|
| <b>show</b>   | Displays the current access point firmware update settings.       |
| <b>set</b>    | Defines the access point firmware update parameters.              |
| <b>update</b> | Executes the firmware update.                                     |
| <b>..</b>     | Goes to the parent menu.  |
| <b>/</b>      | Goes to the root menu.  |
| <b>save</b>   | Saves the current configuration to the access point system flash. |
| <b>quit</b>   | Quits the CLI and exits the current session.                      |

## **admin(system.fw-update)> show**

Displays the current access point firmware update settings.

### **Syntax**

**show** Shows the current system firmware update settings for the access point.

### **Example**

```
admin(system.fw-update)>show
```

```
automatic firmware upgrade      : enable
automatic config upgrade        : enable

firmware filename               : APFW.bin
firmware path                   : /tftpboot/
ftp/tftp server ip address      : 168.197.2.2
ftp user name                   : jsmith
ftp password                    : *****
```

**admin(system.fw-update)> set**

Defines access point firmware update settings and user permissions.

**Syntax**

|            |          |            |   |
|------------|----------|------------|---|
| <b>set</b> | fw-auto  | <mode>     | When enabled, updates device firmware each time the firmware versions are found to be different between the access point and the specified firmware on the remote system.     |
|            | cfg-auto | <mode>     | When enabled, updates device configuration file each time the config file versions are found to be different between the access point and the specified LAN or WAN interface. |
|            | file     | <name>     | Defines the firmware file name (1 to 39 characters).  |
|            | path     | <path>     | Specifies a path for the file (1 to 39 characters)..  |
|            | server   | <ip>       | The IP address for the FTP/TFTP server used for the firmware and/or config file update.   |
|            | user     | <name>     | Specifies a username for FTP server login (1 to 39 characters).   |
|            | passwd   | <password> | Specifies a password for FTP server login (1 to 39 characters). Default is admin123.  |

```
admin(system.fw-update)>set fw-auto enable
admin(system.fw-update)>set cfg-auto enable
admin(system.fw-update)>set file 2.0.0.0-29D
admin(system.fw-update)>set path c:/fw
admin(system.fw-update)>set server 157.235.111.22
admin(system.fw-update)>set user mudskipper
admin(system.fw-update)>set passwd muddy
```



## **admin(system.fw-update)> update**

Executes the access point firmware update over the WAN or LAN port using either ftp or tftp.

### **Syntax**

**update** <mode> Defines the ftp or tftp mode used to conduct the firmware update. Specifies whether the update is executed over the access point's WAN, LAN1 or LAN2 interface <iface>.



#### **NOTE**

---

*The access point must complete the reboot process to successfully update the device firmware, regardless of whether the reboot is conducted using the GUI or CLI interfaces.*

```
admin(system.fw-update)>update ftp
```

## Statistics Commands

### **admin>stats**

Navigates to the access point statistics submenu. The items available under this command are:

|                       |  |
|-----------------------|--|
| <b>show</b>           | Displays access point WLAN, Client, LAN and WAN statistics.            |
| <b>send-cfg-ap</b>    | Sends a config file to another access point within the known AP table. |
| <b>send-cfg-all</b>   | Sends a config file to all access points within the known AP table.    |
| <b>clear</b>          | Clears all statistic counters to zero.                                 |
| <b>flash-all-leds</b> | Starts and stops the flashing of all access point LEDs.                |
| <b>echo</b>           | Defines the parameters for pinging a designated station.               |
| <b>ping</b>           | Initiates a ping test.   |
| <b>..</b>             | Moves to the parent menu.  |
| <b>/</b>              | Goes to the root menu.   |
| <b>save</b>           | Saves the current configuration to system flash.                       |
| <b>quit</b>           | Quits the CLI.   |

## **admin(stats)> show**

Displays access point system information.

### **Syntax**

|             |             |   |
|-------------|-------------|---|
| <b>show</b> | wan         | Displays stats for the access point WAN port.             |
|             | leases      | Displays the leases issued by the access point.           |
|             | lan         | Displays stats for the access point LAN port              |
|             | stp         | Displays LAN Spanning Tree Status                         |
|             | wlan        | Displays WLAN status and statistics summary.              |
|             | s-wlan      | Displays status and statistics for an individual WLAN     |
|             | radio       | Displays a radio statistics transmit and receive summary. |
|             | s-radio     | Displays radio statistics for a single radio              |
|             | retry-hgram | Displays a radio's retry histogram statistics.            |
|             | mu          | Displays all mobile unit (Client) status.                 |
|             | s-mu        | Displays status and statistics for an individual Client.  |
|             | auth-mu     | Displays single Client Authentication statistics.         |
|             | wlap        | Displays Wireless Bridge Statistics statistics summary.   |
|             | s-wlap      | Displays single Wireless Bridge statistics.               |
|             | known-ap    | Displays a Known AP summary.                              |
|             | cpu-mem     | Displays memory and CPU usage statistics.                 |

## **admin(stats)> send-cfg-ap**

Copies the access point's configuration to another access point within the known AP table.

### **Syntax**

|                    |                    |   |
|--------------------|--------------------|---|
| <b>send-cfg-ap</b> | <b>&lt;idx&gt;</b> | Copies the access point's configuration to the access points within the known AP table. Mesh configuration attributes do not get copied using this command and must be configured manually. |
|--------------------|--------------------|---|

### **Example**

```
admin(stats)>send-cfg-ap 2
admin(stats)>
```



#### **NOTE**

---

*The send-cfg-ap command copies all existing configuration parameters except Mesh settings, LAN IP data, WAN IP data and DHCP Server parameter information.*

## **admin(stats)> send-cfg-all**

Copies the access point's configuration to all of the access points within the known AP table.

### **Syntax**

**send-cfg-all**      Copies the access point's configuration to all of the access points within the known AP table.

### **Example**

```
admin(stats)>send-cfg-all  
admin(stats)>
```



#### **NOTE**

---

*The send-cfg-all command copies all existing configuration parameters except Mesh settings, LAN IP data, WAN IP data and DHCP Server parameter information.*

**admin(stats)> clear**

Clears the specified statistics counters to zero to begin new data calculations.

**Syntax**

|              |           |   |
|--------------|-----------|---|
| <b>clear</b> | wan       | Clears WAN statistics counters.   |
|              | lan       | Clears statistics counters for specified LAN index (either clear lan 1 or clear lan 2). |
|              | all-rf    | Clears all RF data.   |
|              | all-wlan  | Clears all WLAN summary information.  |
|              | wlan      | Clears individual WLAN statistic counters.  |
|              | all-radio | Clears access point radio summary information.  |
|              | radio1    | Clears statistics counters specific to radio1.  |
|              | radio2    | Clears statistics counters specific to radio2.  |
|              | all-mu    | Clears all Client statistic counters.   |
|              | mu        | Clears Client statistics counters.  |
|              | known-ap  | Clears Known AP statistic counters.   |

## **admin(stats)> flash-all-leds**

Starts and stops the illumination of a specified access point's LEDs.

### **Syntax**

|                       |          |  |
|-----------------------|----------|--|
| <b>flash-all-leds</b> | <idx>    | Defines the Known AP index number of the target AP to flash. |
|                       | <action> | Starts or stops the flash activity.                          |

### **Example**

```
admin(stats)>
```

```
admin(stats)>flash-all-leds 1 start
```

```
Password *****
```

```
admin(stats)>flash-all-leds 1 stop
```

```
admin(stats)>
```

**admin(stats)> echo**

Defines the echo test values used to conduct a ping test to an associated Client.

**Syntax**

|              |   |
|--------------|---|
| <b>show</b>  | Shows the Mobile Unit Statistics Summary. |
| <b>list</b>  | Defines echo test parameters and result.  |
| <b>set</b>   | Determines echo test packet data.         |
| <b>start</b> | Begins echoing the defined station.       |
| <b>..</b>    | Goes to parent menu.                      |
| <b>/</b>     | Goes to root menu.                        |
| <b>quit</b>  | Quits CLI session.                        |



## **admin(stats.echo)> show**

Shows Mobile Unit Statistics Summary.

### **Syntax**

**show**                Shows Mobile Unit Statistics Summary.

### **Example**

```
admin(stats.echo)>show
```

```
-----  
Idx      IP Address      MAC Address      WLAN      Radio      T-put      ABS      Retries  
-----  
1         192.168.2.0      00:A0F8:72:57:83 demo      11a
```

**admin(stats.echo)> list**

Lists echo test parameters and results.

**Syntax**

**list**                Lists echo test parameters and results.

**Example**

```
admin(stats.echo)>list
```

```
Station Address           : 00A0F8213434
Number of Pings           : 10
Packet Length             : 10
Packet Data (in HEX)      : 55
```

```
admin(stats.echo)>
```

## **admin(stats.echo)> set**

Defines the parameters of the echo test.

### **Syntax**

|            |         |       |  |
|------------|---------|-------|--|
| <b>set</b> | station | <mac> | Defines a Client target MAC address.             |
|            | request | <num> | Sets number of echo packets to transmit (1-539). |
|            | length  | <num> | Determines echo packet length in bytes (1-539).  |
|            | data    | <hex> | Defines the particular packet data.              |

**admin(stats.echo)> start**

Initiates the echo test.

**Syntax**

**start**      Initiates the echo test.

**Example**

```
admin(stats.echo)>start
```

```
admin(stats.echo)>list
```

|                        |   |              |
|------------------------|---|--------------|
| Station Address        | : | 00A0F843AABB |
| Number of Pings        | : | 10           |
| Packet Length          | : | 100          |
| Packet Data (in HEX)   | : | 1            |
|                        |   |              |
| Number of MU Responses | : | 2            |

## **admin(stats)> ping**

Defines the ping test values used to conduct a ping test to an AP with the same ESSID.

### **Syntax**

|             |       |                                     |
|-------------|-------|-------------------------------------|
| <b>ping</b> | show  | Shows Known AP Summary details.     |
|             | list  | Defines ping test packet length.    |
|             | set   | Determines ping test packet data.   |
|             | start | Begins pinging the defined station. |
|             | ..    | Goes to parent menu.                |
|             | /     | Goes to root menu.                  |
|             | quit  | Quits CLI session.                  |

**admin(stats.ping)> show**

Shows Known AP Summary Details.

**Syntax**

**show** Shows Known AP Summary Details.

**Example**

```
admin(stats.ping)>show
```

| Idx | IP Address  | MAC Address      | MUs | KBIOS | Unit Name    |
|-----|-------------|------------------|-----|-------|--------------|
| 1   | 192.168.2.0 | 00:A0F8:72:57:83 | 3   | 0     | access point |

## **admin(stats.ping)> list**

Lists ping test parameters and results.

### **Syntax**

**list**                      Lists ping test parameters and results.

### **Example**

```
admin(stats.ping)>list
```

```
Station Address           : 00A0F8213434
Number of Pings           : 10
Packet Length             : 10
Packet Data (in HEX)      : 55
```

```
admin(stats.ping)>
```

**admin(stats.ping)> set**

Defines the parameters of the ping test.

**Syntax**

|            |         |  |
|------------|---------|--|
| <b>set</b> | station | Defines the AP target MAC address.               |
|            | request | Sets number of ping packets to transmit (1-539). |
|            | length  | Determines ping packet length in bytes (1-539).  |
|            | data    | Defines the particular packet data.              |

**Example**

```
admin(stats.ping)>set station 00A0F843AABB
admin(stats.ping)>set request 10
admin(stats.ping)>set length 100
admin(stats.ping)>set data 1

admin(stats.ping)>
```



## **admin(stats.echo)> start**

Initiates the ping test.

### **Syntax**

**start**                      Initiates the ping test.

### **Example**

```
admin(stats.ping)>start
```

```
admin(stats.ping)>list
```

|                        |   |              |
|------------------------|---|--------------|
| Station Address        | : | 00A0F843AABB |
| Number of Pings        | : | 10           |
| Packet Length          | : | 100          |
| Packet Data (in HEX)   | : | 1            |
|                        |   |              |
| Number of AP Responses | : | 2            |



The management of an adopted AP is conducted by the controller, once the AP connects to an Extreme Networks Summit WM3600 or Summit WM3700 wireless LAN controller and receives its configuration.

An adopted AP provides:

- local 802.11 traffic termination
- local encryption/decryption
- local traffic bridging
- the tunneling of centralized traffic to the wireless controller

An AP's controller connection can be secured using IP/UDP or IPSec depending on whether a secure WAN link from a remote site to the central site already exists.

The controller can be discovered using one of the following mechanisms:

- DHCP
- Controller fully qualified domain name (FQDN)
- Static IP addresses

The benefits of a controller managed AP deployment include:

- *Centralized Configuration Management & Compliance* - Wireless configurations across distributed sites can be centrally managed by the wireless controller or cluster.
- *Controller Link Survivability* - Local WLAN services at a remote sites are unaffected in the case of a controller link failure.
- *Securely extend corporate WLAN's to remote sites* - Small home or office deployments can utilize the feature set of a corporate WLAN from their remote location.
- *Maintain local WLAN's for local applications* - WLANs created and supported locally can be concurrently supported with your existing infrastructure.

## Where to Go From Here

Refer to the following for a further understanding of AP operation:

- [AP Management on page 284](#)
- [Types of Adopted APs on page 284](#)
- [Licensing on page 284](#)
- [Controller Discovery on page 284](#)
- [Securing a Configuration Channel Between Controller and AP on page 285](#)
- [AP WLAN Topology on page 286](#)
- [Securing a Configuration Channel Between Controller and AP on page 285](#)
- [Securing Data Tunnels between the Controller and AP on page 286](#)
- [Managing an AP's Controller Failure on page 287](#)

- [Remote Site Survivability \(RSS\) on page 287](#)
- [Mesh Support on page 287](#)

For an understanding of how support should be configured for the access point and its connected controller, see [“How the AP Receives its Configuration” on page 291](#).

For an overview of how to configure both the access point and controller for basic connectivity and operation, see [“Establishing Controller Managed AP Connectivity” on page 292](#).

## AP Management

An AP can be adopted, configured and managed from the wireless controller.



### NOTE

*Configuration changes made on an AP35XX will not be updated on the controller. To change the AP configuration for an AP35XX make the changes using the controller's interface.*

An AP's wireless configuration can also be configured from the controller. However, non-wireless features (DHCP, NAT, Firewall etc.) cannot be configured from the controller and must be defined using the access point's resident interfaces before its controller adoption or through *Extreme Networks Wireless Management System* (WMS).

## Types of Adopted APs

After adoption, the AP receives its configuration from the controller and starts functioning as an adaptive AP. For ongoing operation, the AP35XX needs to maintain connectivity with the controller. If controller connectivity is lost, the AP35XX continues operating as a stand-alone access point for a period of 3 days before resetting and executing the controller discovery algorithm again.

An AP cannot be converted into a standalone AP35XX through a firmware change. Refer to the AP35XX Hardware/Software Compatibility Matrix within the release notes bundled with the access point firmware.

## Licensing

An adopted AP uses an existing license purchased with a controller. Regardless of how many APs are deployed by a controller, you must ensure the license used by the controller supports the number of radio ports you intend to adopt.

## Controller Discovery

An AP35XX needs to connect to a controller to receive its configuration.

## Auto Discovery using DHCP

Extended Global Options 189, 190, 191, 192 can be used or Embedded Option 43 - Vendor Specific options can be embedded in Option 43 using the vendor class identifier.

|  | Code | Data Type |
|--|------|-----------|
| List of Controller IP addresses<br>(separate by comma, semi-colon, or space delimited)                   | 189  | String    |
| Controller FQDN  | 190  | String    |
| AP35xx Encryption IPsec Passphrase (Hashed) **   | 191  | String    |
| AP35xx controller discovery mode<br>1 = auto discovery enable<br>2 = auto discover enabled (using IPsec) | 192  | String    |

\*\* The AP35XX uses an encryption key to hash passphrases and security keys. To obtain the encryption passphrase, configure an AP35XX with the passphrase and export the configuration file.

```
/
enc-admin-passwd d2
/
// System Configuration
/
system
set name
set loc \0
set email \0
set cc us
/
system
aap-setup
// Adaptive AP menu
set auto-discovery disable
set interface lan1
set name \0
set port 24576
delete all
set enc-passphrase bf0819993a702c39
set ac-keepalive 5
set tunnel-to-switch enable
/
// System-Access menu
system
access
set applet lan 1 enable
set applet wlan 1 enable
set cli lan 1 enable
set ssh lan 1 enable
set snmp lan 1 enable
set applet lan 2 enable
set applet wlan 2 enable
set cli lan 2 enable
```

Encrypted Passphrase to be used in DHCP Option

## Securing a Configuration Channel Between Controller and AP

Once an access point obtains a list of available controllers, it begins connecting to the controller according to the priority list.

The controller is discovered by the access point through several L3 discovery mechanisms even though the controller can be either on the same L2 network as the AP's or on the different network segment (L3). This provides flexibility in wireless network deployment. If the controller is on the access point's

LAN, ensure the LAN subnet is on a secure channel. The AP will connect to the controller and request a configuration.

## AP WLAN Topology

An AP can be deployed in the following WLAN topologies:

- *Extended WLANs* - Extended WLANs are centralized WLANs created on the controller. All wireless client traffics are tunneled to the controller.
- *Independent WLANs* - Independent WLANs are local to an AP and can be configured from the controller. You must specify a WLAN as independent to stop traffic from being forwarded to the controller. All wireless data traffics are locally bridged at the AP. Management traffic is forwarded to the controller.
- *Both* - Extended and independent WLANs are configured from the controller and operate simultaneously.



### NOTE

*For a review of some important considerations impacting the use of extended and independent WLANs within an AP deployment, see [“AP Deployment Considerations”](#) on page 297.*

## Configuration Updates

An AP receives its configuration from the controller initially as part of its adoption sequence. Subsequent configuration changes on the controller are reflected on an AP when applicable.

An AP applies the configuration changes it receives from the controller after 30 seconds from the last received controller configuration message. When the configuration is applied on the AP, the radios shutdown and re-initialize (this process takes less than 2 seconds) forcing associated MUs to be deauthenticated. MUs are quickly able to associate.



### NOTE

*When using a dependant mode AP, be aware that any configuration changes made directly on the AP will be overwritten once the AP is adopted by the controller and the configuration file from the controller is received.*

## Securing Data Tunnels between the Controller and AP

If a secure link (site-to-site VPN) from a remote site to the central location already exists, the AP does not require IPSec be configured for adoption.

For sites with no secure link to the central location, an AP can be configured to use an IPSec tunnel (with AES 256 encryption) for adoption. The tunnel configuration is automatic on the AP side and requires no manual VPN policy be configured. On the controller side, configuration updates are required to adopt the AP using an IPSec tunnel.

To review a sample AP configuration, see [“Sample Controller Configuration File for IPSec and Independent WLAN”](#) on page 298.

## Managing an AP's Controller Failure

In the event of a controller failure, an AP's independent WLAN continues to operate without disruption. The AP attempts to connect to other controllers (if available) in background. Extended WLANs are disabled once controller adoption is lost. When a new controller is discovered and a connection is secured, an extended WLAN is resumed automatically.

If a new controller is located, the AP synchronizes its configuration with the located controller once adopted. If *Remote Site Survivability* (RSS) is disabled, the independent WLAN is also disabled in the event of a controller failure.

## Remote Site Survivability (RSS)

RSS can be used to turn off RF activity on an AP if it loses adoption (connection) to the controller.

| RSS State    | Independent WLANs        | Extended WLANs   |
|--------------|--------------------------|--|
| RSS Enabled  | WLAN continues beaconing | WLAN continues beaconing but AP does allow clients to associate on that WLAN |
| RSS Disabled | WLAN stops beaconing     | WLAN stops beaconing   |

## Mesh Support

An AP can extend existing mesh functionality to a controller managed network. Mesh topology is configured partly through the wireless controller (defining the role of each mesh node) and partly at the mesh AP (defining the connection weight of each backhaul link). APs without a wired connection form a mesh backhaul to a repeater or a wired mesh node and then get adopted to the controller. Mesh nodes with existing wired access get adopted to the controller like a wired AP.

Setting a mesh takes two phases: the first phase is mesh AP staging during which all mesh APs are wired to the controller (L2 or L3) for configuration. The second phase is mesh deployment: disconnect all the remote APs (repeater APs and client bridge APs) to establish wireless backhaul connections. After the deployment, the mesh backhaul topology cannot be modified.

Mesh supported APs apply configuration changes 180 seconds after the last received controller configuration message. When the configuration is applied on the Mesh AP, the radios shutdown and re-initialize (this process takes less than 2 seconds), forcing associated MUs to be deauthenticated and the Mesh link will go down. MUs are able to quickly associate, but the Mesh link will need to be re-established before MUs can pass traffic. This typically takes about 90 to 180 seconds depending on the size of the mesh topology.



### NOTE

*When mesh is employed, the "ap-timeout" value needs to be set to a higher value (for example, 180 seconds) so Mesh APs remain adopted to the controller during the period when the configuration is applied and mesh links are re-established.*

## AP Radius Proxy Support

When an AP is adopted to a controller over a WAN Link, the controller configures the AP for a WLAN with Radius authentication from a Radius server residing at the central site. When the AP gets a Radius

MU associated, it sends the Radius packets on the wired side with its own IP Address as the source IP of the request and the Destination IP Address of the Radius Server. In a local network implementation, the APs, controller and Radius Servers are all on the same LAN and the routing works fine. However, when the AP is adopted over a WAN link, the Radius Server IP Address will be an internal address which is non-routable over the Internet.

To access the Radius server's non-routable IP address over the WAN, you have the option to configure AP Radius Proxying for the WLAN. When this flag is enabled, the AP is reconfigured to send all radius traffic to the controller and the controller does the proxying to the real Radius server to handle authentication. The controller automates the process of handling Radius proxy configuration and client configurations. The controller supports only one real radius server configuration without the presence of realm information. To support multiple radius servers, a realm has to be associated with the real Radius server.

When radius proxying is enabled without specifying a realm, the controller can no longer process requests on the on-board radius server. You cannot authenticate using the on-board Radius server any longer because all authentications done by users without a realm are forwarded to the external radius server, as configured for the WLAN with AP Radius Proxy.



#### NOTE

*The Extreme Networks wireless LAN controllers support AP Radius proxy without specifying realm information. If AP Proxy Radius is enabled without specifying realm information, the onboard Radius server can no longer be used to authenticate users. If Proxy Radius is enabled for a WLAN with realm configured, then the onboard Radius server can perform as usual.*



#### NOTE

*If AP Proxy Radius is configured, the onboard Radius server has to be enabled. By default the onboard Radius server is disabled. To enable the onboard Radius server use the Web UI or issue the "service radius" command in the CLI.*

## Supported AP Topologies

The following AP topologies are supported:

- [Extended WLANs Only](#)
- [Independent WLANs Only](#)
- [Extended WLANs with Independent WLANs](#)
- [Extended VLAN with Mesh Networking](#)

## Topology Deployment Considerations

When reviewing the AP topologies described in the section, be cognizant of the following considerations to optimize the effectiveness of the deployment:

- An AP firmware upgrade will not be performed at the time of adoption from the wireless controller. Instead, the firmware is upgraded using the AP-35xx's firmware update procedure (manually or using the DHCP Auto Update feature).



- There are two LAN interfaces on the AP35xx LAN port: LAN1 and LAN2. By default, LAN1 is the primary LAN connection. LAN2 is only used for tunneled traffic.
- An AP can use its LAN1 interface on the LAN port or WAN interface for adoption. The default gateway interface is set to LAN1. If the WAN Interface is used, explicitly configure WAN as the default gateway interface.
- Extreme Networks recommends using the LAN1 interface for adoption in multi-cell deployments.
- If you have multiple independent WLANs mapped to different VLANs, the AP's LAN1 interface requires trunking be enabled with the correct management and native VLAN IDs configured. Additionally, the AP needs to be connected to a 802.1q trunk port on the wired controller.
- Be aware IPSec Mode supports NAT Traversal (NAT-T).

## Extended WLANs Only

An extended WLAN configuration forces all MU traffic through the controller (tunneled traffic). No wireless traffic is locally bridged at the AP.

Each extended WLAN is mapped to the access point's virtual LAN2 subnet. By default, the access point's LAN2 is not enabled and the default configuration is set to static with IP addresses defined as all zeros. If the extended VLAN option is configured on the controller, the following configuration updates are made automatically:

- The AP's LAN2 subnet becomes enabled
- All extended VLANs are mapped to LAN2.



### NOTE

**MUs on the same WLAN associated to the AP can communicate locally at the AP Level without going through the controller. If this scenario is undesirable, the access point's MU-to-MU disallow option should be enabled.**

## Independent WLANs Only

An independent WLAN configuration forces all MU traffic be bridged locally by the AP. No wireless traffic is tunneled back to the controller. Each extended WLAN is mapped to the access point's LAN1 interface. The only traffic between the controller and the AP are control messages (for example, heartbeats, statistics and configuration updates).

## Extended WLANs with Independent WLANs

An AP can have both extended WLANs and independent WLANs operating in conjunction. When used together, MU traffic from extended WLANs go back to the controller and traffic from independent WLANs is bridged locally by the AP.

All local WLANs are mapped to LAN1, and all extended WLANs are mapped to LAN2.

## Extended VLAN with Mesh Networking

Mesh networking is an extension of the existing wired network. There is no special configuration required, with the exception of setting the mesh and using it within one of the two extended VLAN configurations.



### NOTE

*The mesh backhaul WLAN must be an independent WLAN mapped to LAN2. The controller enforces the mesh WLAN be defined as an independent WLAN by automatically setting the WLAN to independent when backhaul is selected. The AP ensures the backhaul WLAN be put on LAN1.*

## How the AP Receives its Configuration

An AP does not require a separate "local" or "running" configuration. Once adopted, the AP obtains its configuration from the controller. If the AP to controller link fails, it continues to operate using the last valid configuration until its link is re-established and a new configuration is pushed down from the controller. There is no separate file-based configuration stored on the controller.

Only WLAN, VLAN extension and radio configuration items are defined for the AP by its connected controller. None of the other access point configuration items (RADIUS, DHCP, NAT, Firewall etc.) are configurable from the connected controller.

After the AP downloads a configuration file from the controller, it obtains the version number of the image it should be running. The controller does not have the capacity to hold the access point's firmware image and configuration. The access point image must be downloaded using a means outside the controller. If there is still an image version mismatch between what the controller expects and what the AP is running, the controller will deny adoption.



### NOTE

*When configuring wireless settings for APs, all configuration must be done through the controller and not from the AP management console. Making changes directly in the AP management console can lead to unstable operation of the AP.*

## AP Adoption Prerequisites

Adopting an AP3510 model access point requires:

- The appropriate controller licenses providing AP functionality on the controller.
- The correct password to authenticate and connect the AP to the controller.

## Configuring the AP for Adoption by the Controller

To configure an AP for controller adoption:

- 1 An AP needs to find and connect to the controller. To ensure this connection:
  - Configure the controller's IP address on the AP.
  - Provide the controller IP address using DHCP option 189 on a DHCP server. The IP address is a comma delimited string of IP addresses. For example "157.235.94.91, 10.10.10.19". There can be a maximum of 12 IP addresses.
  - Configure the controller's FQDN on the AP. The AP can use this to resolve the IP address of the controller.
- 2 Use the controller's secret password on the AP for the controller to authenticate it.

To avoid a lengthy broken connection with the controller, Extreme Networks recommends generating an SNMP trap when the AP loses adoption with the controller.

**NOTE**

*For additional information (in greater detail) on the AP configuration activities described above, see [“AP Configuration”](#) on page 292.*

## Configuring the Controller for AP Adoption

The tasks described below are configured on an Extreme Networks wireless LAN controller.

To adopt an AP on a controller:

- 1 Ensure enough licenses are available on the controller to adopt the required number of APs.
- 2 As soon as the AP displays in the adopted list:  
Adjust each AP's radio configuration as required. This includes WLAN-radio mappings and radio parameters. WLAN-VLAN mappings and WLAN parameters are global and cannot be defined on a per radio basis. WLANs can be assigned to a radio. Optionally, configure WLANs as independent and assign to APs as needed.
- 3 Configure each VPN tunnel with the VLANs to be extended to it.  
If you do not attach the target VLAN, no data will be forwarded to the AP, only control traffic required to adopt and configure the AP.

**NOTE**

*For additional information (in greater detail) on the controller configuration activities described above, see [“Controller Configuration”](#) on page 293.*

## Establishing Controller Managed AP Connectivity

This section defines the activities required to configure basic AP connectivity with the controller. In establishing a basic AP connection, both the access point and controller require modifications to their respective default configurations. For more information, see:

- [AP Configuration on page 292](#)
- [Types of Adopted APs](#)

**NOTE**

*Refer to [“AP Deployment Considerations”](#) on page 297 for usage and deployment caveats that should be considered before defining the AP configuration. Refer to [“Sample Controller Configuration File for IPSec and Independent WLAN”](#) on page 298 if planning to deploy an AP configuration using IPSec VPN and an extended WLAN*

## AP Configuration

An AP can be adopted using a configuration file pushed to the AP or adopted using DHCP options. Each of these adoption techniques is described in the sections that follow.

## Adopting an AP Using a Configuration File

To adopt an AP using a configuration file:

- 4 Define the AP controller connection parameters.
- 5 Export the AP's configuration to a secure location.

Either import the configuration manually to other APs or the same AP later (if you elect to default its configuration). Use DHCP option 186 and 187 to force a download of the configuration file during startup (when it receives a DHCP offer).



### NOTE

*When an AP is adopted over an IP Sec Tunnel you cannot export the configuration file to a system on the other side of the IP Sec Tunnel. You may still export the configuration file to a system local to the AP.*

## Adopting an AP Using DHCP Options

An AP can be adopted to a wireless controller by providing the following options in the DHCP Offer:

| Option | Data Type | Value   |
|--------|-----------|---|
| 189    | String    | <Controller IP Address or Range of IP addresses separated by [, ; <space>]> |
| 190    | String    | <Fully qualified Domain Name for the Wireless Controller>                   |
| 191    | String    | <Hashed IPSec Passkey - configure on 1 AP and export to get hashed key>     |
| 192    | String    | <Value of "1" denotes Non-IPSec Mode and "2" denotes IPSec Mode>            |



### NOTE

*Options 189 and 192 are mandatory to trigger adoption using DHCP options. Option 189 alone won't work. These options can be embedded in Vendor Specific Option 43 and sent in the DHCP Offer.*

## Controller Configuration

An Extreme Networks wireless LAN controller requires an explicit configuration to adopt an AP (if IPSec is not being used for adoption).

Disable the controller's *Adopt unconfigured radios automatically* option and manually add APs requiring adoption, or leave as default. In default mode, any AP adoption request is honored until the current controller license limit is reached.

To disable automatic adoption on the controller:

- 1 Select *Network > Access Point Radios* from the controller main menu tree.
- 2 Select the *Configuration* tab (should be displayed by default) and click the *Global Settings* button.

Network > Access Port Radios > Global

**Global**

Controller Adoption Preference ID  (1 - 65535)

☐ Adopt unconfigured radios automatically

☐ Voice Call Admission Control

Primary WIPS Server Address

Secondary WIPS Server Address

Status:

- 3 Ensure the *Adopt unconfigured radios automatically* option is NOT selected.  
When disabled, there is no automatic adoption of non-configured radios on the network. Additionally, default radio settings will NOT be applied to access points when automatically adopted.

### NOTE

For IPSec deployments, refer to [“Sample Controller Configuration File for IPSec and Independent WLAN” on page 298](#) and take note of the CLI commands in red and associated comments in green.

Any WLAN configured on the controller becomes an extended WLAN by default for an AP.

- 4 Select *Network > Wireless LANs* from the controller main menu tree.
- 5 Select the target WLAN you would like to use for AP support from those displayed and click the *Edit* button.
- 6 Select the *Independent Mode* checkbox.  
Selecting the checkbox designates the WLAN as independent and prevents traffic from being forwarded to the controller. Independent WLANs behave like WLANs as used on a standalone access point. Leave this option unselected (as is by default) to keep this WLAN an extended WLAN (a typical centralized WLAN created on the controller).

Network > Wireless LANs > Edit

**Edit** WLAN7

---

**Configuration**

ESSID  Description

☒ Independent Mode ☐ Client Bridge Backhaul

VLAN ID  ☐ Dynamic Assignment

---

**Authentication**

☐ 802.1X EAP   
☐ Kerberos   
☐ Hotspot   
☐ MAC Authentication   
☒ No Authentication

**Encryption**

☐ WEP 64   
☐ WEP 128   
☐ KeyGuard   
☐ WPA/WPA2-TKIP   
☐ WPA2-CCMP

---

**Advanced**

Accounting Mode  MU to MU Traffic   
☒ Answer Broadcast ESS MU Idle Time  seconds  
☐ Use Voice Prioritization Access Category   
☐ Enable SVP MCast Addr 1   
☐ Secure Beacon MCast Addr 2   
 QoS Weight  NAC Mode

---

Status:



#### NOTE

Additionally, a WLAN can be defined as independent using the "wlan <index> independent" command from the config-wireless context.



#### NOTE

Avoid mapping independent or extended WLANs to VLANs on the controller's ge port.

Once an AP is adopted by the controller, it displays within the controller's *Access Point Radios* screen (under the Network parent menu item) as an AP3510 or AP3550.



## AP Deployment Considerations

Before deploying your controller/AP configuration, refer to the following usage caveats to optimize its effectiveness:

- Extended WLANs are mapped to the AP's LAN2 interface and all independent WLANs are mapped to the AP's LAN1 Interface.
- If deploying multiple independent WLANs mapped to different VLANs, ensure the AP's LAN1 interface is connected to a trunk port on the Layer 2/Layer 3 controller and appropriate management and native VLANs are configured.
- The WLAN used for mesh backhaul must always be an independent WLAN.
- The controller configures an AP. If manually changing wireless settings on the AP, they are not updated on the controller. It's a one way configuration, from the controller to the AP.
- Adopting an AP always requires a router between the AP and the controller.
- An adopted AP can be used behind a NAT.
- An AP uses UDP port 24576 for control frames and UDP port 24577 for data frames.
- Multiple VLANs per WLAN, Layer 3 mobility, dynamic VLAN assignment, NAC, rogue AP, MU locationing, hotspot on extended WLAN are some of the important wireless features not supported in an AP supported deployment.

## Sample Controller Configuration File for IPSec and Independent WLAN

The following constitutes a sample controller configuration file supporting an AP IPSec with Independent WLAN configuration. Please note new AP specific CLI commands in **red** and relevant comments in **blue**.

The sample output is as follows:

```
!
! configuration of WM3600
!
!
aaa authentication login default none
service prompt crash-info
!
hostname WM3600-1
!
username admin password 1 8e67bb26b358e2ed20fe552ed6fb832f397a507d
username admin privilege superuser
username operator password 1 fe96dd39756ac41b74283a9292652d366d73931f
!
!
To configure the ACL to be used in the CRYPTO MAP
!
ip access-list extended AAP-ACL permit ip host 10.10.10.250 any rule-precedence 20
!
spanning-tree mst cisco-interoperability enable
spanning-tree mst config
name My Name
!
country-code us
logging buffered 4
logging console 7
logging host 157.235.92.97
logging syslog 7
snmp-server sysname WM3600-1
snmp-server manager v2
snmp-server manager v3
snmp-server user snmptrap v3 encrypted auth md5 0x7be2cb56f6060226f15974c936e2739b
snmp-server user snmpmanager v3 encrypted auth md5 0x7be2cb56f6060226f15974c936e2739b
snmp-server user snmpoperator v3 encrypted auth md5 0x49c451c7c6893ffcede0491bbd0a12c4
!
To configure the passkey for a Remote VPN Peer - 255.255.255.255 denotes all AAPs.
12345678 is the default passkey. If you change on the AAP, change here as well.
!
crypto isakmp key 0 12345678 address 255.255.255.255
!
ip http server
ip http secure-trustpoint default-trustpoint
ip http secure-server
ip ssh
no service pm sys-restart
timezone America/Los_Angeles
license AP
```



```

radio add 4 00-15-70-00-79-12 11a aap3510
radio 4 bss 1 5
radio 4 bss 2 6
radio 4 channel-power indoor 48 4
radio 4 rss enable
radio 4 client-bridge bridge-select-mode auto
radio 4 client-bridge ssid Mesh
radio 4 client-bridge mesh-timeout 0
radio 4 client-bridge enable
radio default-11a rss enable
radio default-11bg rss enable
radio default-11b rss enable
no ap-ip default-ap controller-ip
!
radius-server local
!
To create an IPSEC Transform Set
!
crypto ipsec transform-set AAP-TFSET esp-aes-256 esp-sha-hmac mode tunnel
!
To create a Crypto Map, add a remote peer, set the mode, add a ACL rule to match and
transform and set to the Crypto Map
!
crypto map AAP-CRYPTOMAP 10 ipsec-isakmp
set peer 255.255.255.255
set mode aggressive
match address AAP-ACL
set transform-set AAP-TFSET
!
interface ge1
controllerport mode trunk
controllerport trunk native vlan 1
controllerport trunk allowed vlan none
controllerport trunk allowed vlan add 1-9,100,110,120,130,140,150,160,170,
controllerport trunk allowed vlan add 180,190,200,210,220,230,240,250,
static-channel-group 1
!
interface ge2
controllerport access vlan 1
!
interface ge3
controllerport mode trunk
controllerport trunk native vlan 1
controllerport trunk allowed vlan none
controllerport trunk allowed vlan add 1-9,100,110,120,130,140,150,160,170,
controllerport trunk allowed vlan add 180,190,200,210,220,230,240,250,
static-channel-group 1
!
interface ge4
controllerport access vlan 1
!
interface me1
ip address dhcp
!
interface sa1
controllerport mode trunk

```

```
controllerport trunk native vlan 1
controllerport trunk allowed vlan none
controllerport trunk allowed vlan add 1-9,100,110,120,130,140,150,160,170,
controllerport trunk allowed vlan add 180,190,200,210,220,230,240,250,
!
!
!
!
interface vlan1
ip address dhcp
!
To attach a Crypto Map to a VLAN Interface
!
crypto map AAP-CRYPTOMAP
!
sole
!
ip route 157.235.0.0/16 157.235.92.2
ip route 172.0.0.0/8 157.235.92.2
!
ntp server 10.10.10.100 prefer version 3
line con 0
line vty 0 24
!
end
```



The following list of countries and their country codes is useful when using the access point configuration file, CLI or the MIB to configure the access point:

| <b>Country</b>      | <b>Code</b> | <b>Country</b>       | <b>Code</b> |
|---------------------|-------------|----------------------|-------------|
| Argentina           | AR          | Mexico               | MX          |
| Australia           | AU          | Montenegro           | ME          |
| Austria             | AT          | Morocco              | MA          |
| Bahamas             | BS          | Netherlands          | NL          |
| Bahrain             | BH          | Netherlands Antilles | AN          |
| Barbados            | BB          | New Zealand          | NZ          |
| Belarus             | BY          | Nicaragua            | NI          |
| Bermuda             | BM          | Norfolk Island       | NF          |
| Belgium             | BE          | Norway               | NO          |
| Bolivia             | BO          | Oman                 | OM          |
| Botswana            | BW          | Panama               | PA          |
| Botznia-Herzegovina | BA          | Pakistan             | PK          |
| Brazil              | BR          | Paraguay             | PY          |
| Bulgaria            | BG          | Peru                 | PE          |
| Canada              | CA          | Philippines          | PH          |
| Cayman Islands      | KY          | Poland               | PL          |
| Chile               | CL          | Portugal             | PT          |
| China               | CN          | Puerto Rico          | PR          |
| Christmas Islands   | CX          | Qatar                | QA          |
| Colombia            | CO          | Romania              | RO          |
| Costa Rica          | CR          | Russian Federation   | RU          |
| Croatia             | HR          | Saudi Arabia         | SA          |
| Cypress             | CY          | Serbia               | RS          |
| Czech Rep.          | CZ          | Singapore            | SG          |
| Denmark             | DK          | Slovak Republic      | SK          |
| Dominican Republic  | DO          | Slovenia             | SI          |
| Ecuador             | EC          | South Africa         | ZA          |
| El Salvador         | SV          | South Korea          | KR          |

|                  |    |                          |    |
|------------------|----|--------------------------|----|
| Estonia          | EE | Spain                    | ES |
| Egypt            | EG | Sri Lanka                | LK |
| Falkland Islands | FK | Sweden                   | SE |
| Finland          | FI | Switzerland              | CH |
| France           | FR | Taiwan                   | TW |
| Germany          | DE | Thailand                 | TH |
| Greece           | GR | Trinidad and Tobago      | TT |
| Guam             | GU | Turkey                   | TR |
| Guatemala        | GT | Ukraine                  | UA |
| Guinea           | GN | UAE                      | AE |
| Haiti            | HT | United Kingdom           | GB |
| Honduras         | HN | USA                      | US |
| Hong Kong        | HK | Uruguay                  | UY |
| Hungary          | HU | Virgin Islands (British) | VG |
| Iceland          | IS | Virgin Islands (US)      | VI |
| India            | IN | Vietnam                  | VN |
| Indonesia        | ID | Venezuela                | VE |
| Ireland          | IE |                          |    |
| Israel           | IL |                          |    |
| Italy            | IT |                          |    |
| Jamaica          | JM |                          |    |
| Japan            | JP |                          |    |
| Jordan           | JO |                          |    |
| Kazakhstan       | KZ |                          |    |
| Kuwait           | KW |                          |    |
| Latvia           | LV |                          |    |
| Lebanon          | LB |                          |    |
| Liechtenstein    | LI |                          |    |
| Lithuania        | LT |                          |    |
| Luxembourg       | LU |                          |    |
| Macau            | MO |                          |    |
| Macedonia        | MK |                          |    |
| Malaysia         | MY |                          |    |
| Malta            | MT |                          |    |
| Martinique       | MQ |                          |    |







**NOTE**

*Services can be purchased from Extreme Networks or through one of its channel partners. If you are an end-user who has purchased service through an Extreme Networks channel partner, please contact your partner first for support.*

Extreme Networks Technical Assistance Centers (TAC) provide 24x7x365 worldwide coverage. These centers are the focal point of contact for post-sales technical and network-related questions or issues. TAC will create a Service Request (SR) number and manage all aspects of the SR until it is resolved. For a complete guide to customer support, see the *Technical Assistance Center User Guide* at:

[www.extremenetworks.com/go/TACUserGuide](http://www.extremenetworks.com/go/TACUserGuide)

The Extreme Networks eSupport website provides the latest information on Extreme Networks products, including the latest Release Notes, troubleshooting, downloadable updates or patches as appropriate, and other useful information and resources. Directions for contacting the Extreme Networks Technical Assistance Centers are also available from the eSupport website at:

<https://esupport.extremenetworks.com>

## Registration

If you have not already registered this product with Extreme Networks, you can register on the Extreme Networks website at:

<http://www.extremenetworks.com/go/productregistration>

## Documentation

Check for the latest versions of documentation on the Extreme Networks documentation website at:

<http://www.extremenetworks.com/go/documentation>

